

SUMMER 1984

An Argus Specialist Publication

Personal SOFTWARE

£3.99

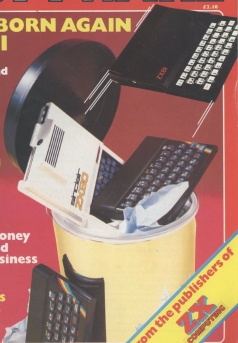
THE BORN AGAIN ZX81

Games and
teasers
to drive
you wild

Get
educated
and
have fun

Make money
and mind
your business

Graphics
come to
life



From the publishers of
ZX
COMPUTING

A NEW, IMPROVED ZX81 KEYBOARD AT THE SAME OLD PRICE. £9.95.



There's only one thing wrong with the ZX81, its keyboard.

Or rather its lack of one.

Since it's flat your fingers don't feel as if there's any response to the pressure put on the keys.

ZX81 KEYS

FILESIXTY KEYS

In other words, you're not quite sure which keys you've pressed until the screen actually tells you.

Our new, improved push button keyboard changes all that.

It matches the ZX81 perfectly. And the keys give a real calculator-type feel.

To set it up all you have to do is peel off the adhesive backing and stick it on top of the ZX81 touchpad.

Because no tampering or soldering is involved the guarantee is not affected. And it will last for up to 3½ million operations.

But our keyboard doesn't just come loaded with features. With it comes a separate overlay and a set of coloured stick-on labels to make game playing easier.

It's yours for the original price of £9.95.

Whichever way you look at it, we think you'll agree that it's a keyboard that's quite outstanding.

Orders to Filesixty Ltd., FREEPOST, London W9 2BR,
Cheques/P.O. made payable to Filesixty Ltd.
Please send me _____ (type keyboard) at £9.95 each
(including VAT and P&P).

Total £

(please continue)

Name

Address

FILESIXTY

Filesixty Ltd., 25 Clapton Road, London W9 2BR, England. Tel: 01-289 3056 Telex: 268 048 EXT LON G 4082



CONTENTS

Editorial & Advertisement Office
No 1 Golden Square, London W1R 3AB
Telephone: 01-437 0826 Telex: 8811896

Volume 3 No 1 Summer 1984

Editor: Wendy Palmer
Sales Executive:
Rory Sinclair
Advertisement Manager:
Rory Sinclair
Advertisement Copy
Control:
Ann McDermott
Managing Editor:
Ron Harris
Chief Executive: T J Coussé
Origination and design by
RM Design & Print

Personal Software is a quarterly
magazine concentrating on the third
family in the market, Sinclair, Spectrum
and Amstrad.

Subscription by Air: Free, July &
September 1984, £2.50, £2.50
London (UK), £3.00, £3.00
Distribution by: Magazine Services &
New Ltd, York, Yorkshire, York

The contents of this publication
including all articles, designs, plans,
drawings and programs are the
copyright and other intellectual
property rights of the authors. All
rights are reserved. No part of this
publication may be reproduced,
stored in a retrieval system, or
transmitted in any form or by any
means, electronic, mechanical, or
otherwise, without prior written
consent of the Copyright and other
rights holders. © 1984 Amstrad
Specialist Publications Limited.

Subscription Rates: UK £2.50/£3.00
per annum including postage.
Foreign rates and other rates upon
application to Personal Software
Subscription, Manor Ltd, 100
High Street, The Malvern, Hereford
WR11 2JH, UK.

Prelude	4	User Character Set	48
Inexticible	5	All sorts of characters on the ZX81	
Try playing hard to get — the aliens are out to get you!		The Sharp End	48
Motor Madness	6	Three great little games to get you going!	
Enter the Grand Prix and drive yourself mad.		Monster II	50
Caterpillar 2	8	Try to escape from the monsters.	
Watch out you don't become addicted!		The Right Chord	50
Crazy Ballrooming	10	Tricky guitar chords made easy.	
Fly high and fast and watch out for low-flying aircraft.		Kooky Satellite	55
Talking Turkey	12	Here is on the money.	
Let your mouse create a new dictionary.		On Target	55
Worms	13	Two great programs — Target and Missile Launcher.	
Surround yourself in machine code.		Smoggy's Lair	58
Rockygnosis	16	Can you destroy the dragon's lair?	
See if you can hear off before the computer.		Naval Monoceros	62
3D Mover	20	Get your feet in action.	
Create three-dimensional objects in 10K.		House of Horror	68
Just another Brick	24	I hope you're not heart-hearted.	
Build yourself up to playing games.		Drop Out	68
Number Tumbler	27	A ZX81 version of the old arcade game.	
Is your number up? Read on and find out.		Maths Maze	70
3D Cubes	28	A great game with an educational twist.	
Three-dimensional representations of cubes on the ZX81.		Nought But A Game	70
Grand Prix Driver	31	Don't get cross at the noughts.	
Stop yourself in and prepare for the race of your life.		Peeper	76
Equation Solver	36	A paraded program, so watch out!	
Solve equations using the bisection method.		Five Card Trick	78
Minesfield	39	A fabulous implementation of the card game, Fortoon.	
You really must watch your step.		Profit! Prophet	80
Death Race	41	Services start to help you get to grips with your business.	
Drive your ZX81 to the limit!		Stock Control	84
Towers of Hanoi	42	Keep efficient control of your stock.	
The 3281 solution to this age-old puzzle.		Air Raid	86
Shopping Trial	44	Run for cover and have a smashing time!	
Head 'em up and move 'em out!		Cooling Graphics	90
User Character Set	48	Enlarge your character size with this utility.	
All sorts of characters on the ZX81		Spinning Into Graphics	96
The Sharp End	48	Become artistic with a spinograph program.	
Three great little games to get you going!		Shooter	98
Monster II	50	Simple but fun to play.	
Try to escape from the monsters.			
The Right Chord	50		
Tricky guitar chords made easy.			
Kooky Satellite	55		
Here is on the money.			
On Target	55		
Two great programs — Target and Missile Launcher.			
Smoggy's Lair	58		
Can you destroy the dragon's lair?			
Naval Monoceros	62		
Get your feet in action.			
House of Horror	68		
I hope you're not heart-hearted.			
Drop Out	68		
A ZX81 version of the old arcade game.			
Maths Maze	70		
A great game with an educational twist.			
Nought But A Game	70		
Don't get cross at the noughts.			
Peeper	76		
A paraded program, so watch out!			
Five Card Trick	78		
A fabulous implementation of the card game, Fortoon.			
Profit! Prophet	80		
Services start to help you get to grips with your business.			
Stock Control	84		
Keep efficient control of your stock.			
Air Raid	86		
Run for cover and have a smashing time!			
Cooling Graphics	90		
Enlarge your character size with this utility.			
Spinning Into Graphics	96		
Become artistic with a spinograph program.			
Shooter	98		
Simple but fun to play.			

Credits: Our grateful thanks to the following individuals for their assistance in the
production of this issue: Mark, Gary, Neil, Neil, Nigel, Tom, William, Adrian, Sam, Fiona
and others almost too numerous to mention.

Notes: Sinclair, ZX, ZX Spectrum, ZX Microdrive, ZX Interface are all registered trademarks
of Sinclair Research Ltd.

PREFACE

What's this, I hear someone ask — a magazine for the Z801? But surely the Z801 has had its day and is now no more than a memory? Well, if you are sitting comfortably I'll try to explain...

At a rough estimate, there are a million Z801s out there in computer-kind, and even allowing for those people who bought one as a 'one-day wonder' or have 'progressed' to a more sophisticated microcomputer, there must be an awfully large number of micro owners out there who regularly plug in their Z801. But have you tried to find some exciting new software lately in your local high street store for the Z801? I'd like to hear from anybody that has had recent success in this venture!

It would appear that the software manufacturers have decided that the Z801 is no longer a viable launchpad for new programs and now devote their time, money and effort to the more profitable machines — and who suffers most? You've guessed it — the multitude of Z801 owners. And there, dear reader, is where we have stepped in.

This issue of *Personal Software* is intended to fill the gap left by the commercial suppliers. Most of the programs to be found in these pages were originally printed in our sister magazine, *ZX Computing*. The game of Backgammon appeared first in *Computing Today*. Since they first appeared in print, we have carefully checked them to make sure they contain no errors, and so if you experience any difficulty in getting any of the programs to work, please check them again. If you are still having problems you should write to *Personal Software* stating clearly what problem you have encountered and enclosing a stamped self-addressed envelope. We regret that we cannot answer telephone enquiries.


```

140 LET C=C+INKEY$="S":INKEY$="S"
150 IF PEEK (C+33)=CODE$ THEN
EN GOTO 400
160 POKE C+33,CODE$
170 GOTO 180
180 IF PEEK (C+33)=CODE$ THEN GOTO 5
80
400 IF PEEK (C+33)=CODE$ THEN
R LET C=C+8
410 GOTO 180
500 POKE C,CODE$
610 FOR F=1 TO 4
620 POKE C+F,CODE$
630 POKE C+F,CODE$
640 NEXT F
650 FOR F=1 TO 4
660 POKE C+F,CODE$
670 POKE C+F,CODE$
680 NEXT F
690 FOR F=1 TO 4
700 POKE C+F,CODE$
710 POKE C+F,CODE$
720 FOR U=1 TO 8
730 NEXT U
740 NEXT F
750 PRINT AT 3,5:
760 FOR F=1 TO 80
770 NEXT F
780 CLS
790 POINT "SCORE":50
800 IF 50<=R101 THEN GOTO 910
810 PRINT H: 10.0: "YOU HAVE BE
THIRD IN SCORE IN THE TOP TEN"
820 PRINT AT 20,0: "PLEASE INPUT
YOUR NAME"
830 INPUT A$10
840 LET R102=A$
850 LET R=0
860 FOR I=1 TO 11
870 IF A$(I)=R101 THEN GOTO 76
880 GOTO 890
890 LET T=A$(I)
900 LET C=A$(I)
910 LET R102=R101+1
920 LET A$(I)=R101+1
930 LET R102=R102
940 LET R=R+1
950 NEXT I
960 IF R=0 THEN GOTO 720
970 CLS
980 FOR F=10 TO 3 STEP -1
990 PRINT A$(F)
1000 NEXT F
1010 IF INKEY$="" THEN GOTO 950
1020 CLS
1030 GOTO 5
1040 STOP
1050 FOR F=1 TO 10
1060 LET A$(F)="TTTTT"
1070 LET R101=R
1080 NEXT F
1090 PRINT
1100 PRINT "YOU ARE A RACING ON
TOP NOW"
1110 PRINT "YOU HAVE TO RACE TH
E CHRIS"
1120 PRINT "OCCASIONALLY THERE WI
LL BE ICE ON"
1130 PRINT "THE ROAD AND YOU
WILL TEND"
1140 PRINT "TO SKID ON IT SO BEU
"
1150 PRINT
1160 PRINT
1170 PRINT
1180 IF INKEY$="" THEN GOTO 1130
1190 CLS
1200 RETURN

```


CRAZY BALLOONING

Take part in an aerial assault course.

This is an adaptation of the popular arcade game for the Z80. You have to guide the balloon around the course of prickly asterisks to reach the area marked "GOAL" in the top left-hand corner of the maze.

You start at the bottom right-hand corner of the course and the controls you should use to guide your balloon are shown in Fig. 1. Obviously, the idea of the game is to try and get your balloon to the goal without it hitting the asterisks which, in case you bubble!

PROGRAM DESCRIPTION

Below is a brief breakdown of the program.

You can always take a short cut through one of the two gates in the asterisk wall — but take care, this is not always the easiest route!



Lines 1-65	Initialisation.
Lines 66-185	What key has been pressed?
Lines 186-240	What has the balloon hit?
Lines 241-300	Swing routine.
Lines 301-520	The 'balloon burst' and 'high score' routines.
Lines 560-630	The goal routine.
Lines 631-650	The course at the start of the game.
Lines 1000-1230	The routine to set up the screen.
Lines 2000-2200	The routine containing the instruction.
Lines 3000-3300	The movement routine.

```

1  REM ***** BALLOONS *****
2  LET S=0
3  LET H=0
4  LET D=0
5  LET M=0
6  LET N=0
7  LET P=0
8  LET Q=0
9  LET R=0
10 LET S=0
11 LET H=0
12 LET D=0
13 LET M=0
14 LET N=0
15 LET P=0
16 LET Q=0
17 LET R=0
18 LET S=0
19 LET H=0
20 LET D=0
21 LET M=0
22 LET N=0
23 LET P=0
24 LET Q=0
25 LET R=0
26 LET S=0
27 LET H=0
28 LET D=0
29 LET M=0
30 LET N=0
31 LET P=0
32 LET Q=0
33 LET R=0
34 LET S=0
35 LET H=0
36 LET D=0
37 LET M=0
38 LET N=0
39 LET P=0
40 LET Q=0
41 LET R=0
42 LET S=0
43 LET H=0
44 LET D=0
45 LET M=0
46 LET N=0
47 LET P=0
48 LET Q=0
49 LET R=0
50 LET S=0
51 LET H=0
52 LET D=0
53 LET M=0
54 LET N=0
55 LET P=0
56 LET Q=0
57 LET R=0
58 LET S=0
59 LET H=0
60 LET D=0
61 LET M=0
62 LET N=0
63 LET P=0
64 LET Q=0
65 LET R=0
66 LET S=0
67 LET H=0
68 LET D=0
69 LET M=0
70 LET N=0
71 LET P=0
72 LET Q=0
73 LET R=0
74 LET S=0
75 LET H=0
76 LET D=0
77 LET M=0
78 LET N=0
79 LET P=0
80 LET Q=0
81 LET R=0
82 LET S=0
83 LET H=0
84 LET D=0
85 LET M=0
86 LET N=0
87 LET P=0
88 LET Q=0
89 LET R=0
90 LET S=0
91 LET H=0
92 LET D=0
93 LET M=0
94 LET N=0
95 LET P=0
96 LET Q=0
97 LET R=0
98 LET S=0
99 LET H=0
100 LET D=0
101 LET M=0
102 LET N=0
103 LET P=0
104 LET Q=0
105 LET R=0
106 LET S=0
107 LET H=0
108 LET D=0
109 LET M=0
110 LET N=0
111 LET P=0
112 LET Q=0
113 LET R=0
114 LET S=0
115 LET H=0
116 LET D=0
117 LET M=0
118 LET N=0
119 LET P=0
120 LET Q=0
121 LET R=0
122 LET S=0
123 LET H=0
124 LET D=0
125 LET M=0
126 LET N=0
127 LET P=0
128 LET Q=0
129 LET R=0
130 LET S=0
131 LET H=0
132 LET D=0
133 LET M=0
134 LET N=0
135 LET P=0
136 LET Q=0
137 LET R=0
138 LET S=0
139 LET H=0
140 LET D=0
141 LET M=0
142 LET N=0
143 LET P=0
144 LET Q=0
145 LET R=0
146 LET S=0
147 LET H=0
148 LET D=0
149 LET M=0
150 LET N=0
151 LET P=0
152 LET Q=0
153 LET R=0
154 LET S=0
155 LET H=0
156 LET D=0
157 LET M=0
158 LET N=0
159 LET P=0
160 LET Q=0
161 LET R=0
162 LET S=0
163 LET H=0
164 LET D=0
165 LET M=0
166 LET N=0
167 LET P=0
168 LET Q=0
169 LET R=0
170 LET S=0
171 LET H=0
172 LET D=0
173 LET M=0
174 LET N=0
175 LET P=0
176 LET Q=0
177 LET R=0
178 LET S=0
179 LET H=0
180 LET D=0
181 LET M=0
182 LET N=0
183 LET P=0
184 LET Q=0
185 LET R=0
186 LET S=0
187 LET H=0
188 LET D=0
189 LET M=0
190 LET N=0
191 LET P=0
192 LET Q=0
193 LET R=0
194 LET S=0
195 LET H=0
196 LET D=0
197 LET M=0
198 LET N=0
199 LET P=0
200 LET Q=0
201 LET R=0
202 LET S=0
203 LET H=0
204 LET D=0
205 LET M=0
206 LET N=0
207 LET P=0
208 LET Q=0
209 LET R=0
210 LET S=0
211 LET H=0
212 LET D=0
213 LET M=0
214 LET N=0
215 LET P=0
216 LET Q=0
217 LET R=0
218 LET S=0
219 LET H=0
220 LET D=0
221 LET M=0
222 LET N=0
223 LET P=0
224 LET Q=0
225 LET R=0
226 LET S=0
227 LET H=0
228 LET D=0
229 LET M=0
230 LET N=0
231 LET P=0
232 LET Q=0
233 LET R=0
234 LET S=0
235 LET H=0
236 LET D=0
237 LET M=0
238 LET N=0
239 LET P=0
240 LET Q=0
241 LET R=0
242 LET S=0
243 LET H=0
244 LET D=0
245 LET M=0
246 LET N=0
247 LET P=0
248 LET Q=0
249 LET R=0
250 LET S=0
251 LET H=0
252 LET D=0
253 LET M=0
254 LET N=0
255 LET P=0
256 LET Q=0
257 LET R=0
258 LET S=0
259 LET H=0
260 LET D=0
261 LET M=0
262 LET N=0
263 LET P=0
264 LET Q=0
265 LET R=0
266 LET S=0
267 LET H=0
268 LET D=0
269 LET M=0
270 LET N=0
271 LET P=0
272 LET Q=0
273 LET R=0
274 LET S=0
275 LET H=0
276 LET D=0
277 LET M=0
278 LET N=0
279 LET P=0
280 LET Q=0
281 LET R=0
282 LET S=0
283 LET H=0
284 LET D=0
285 LET M=0
286 LET N=0
287 LET P=0
288 LET Q=0
289 LET R=0
290 LET S=0
291 LET H=0
292 LET D=0
293 LET M=0
294 LET N=0
295 LET P=0
296 LET Q=0
297 LET R=0
298 LET S=0
299 LET H=0
300 LET D=0
301 LET M=0
302 LET N=0
303 LET P=0
304 LET Q=0
305 LET R=0
306 LET S=0
307 LET H=0
308 LET D=0
309 LET M=0
310 LET N=0
311 LET P=0
312 LET Q=0
313 LET R=0
314 LET S=0
315 LET H=0
316 LET D=0
317 LET M=0
318 LET N=0
319 LET P=0
320 LET Q=0
321 LET R=0
322 LET S=0
323 LET H=0
324 LET D=0
325 LET M=0
326 LET N=0
327 LET P=0
328 LET Q=0
329 LET R=0
330 LET S=0
331 LET H=0
332 LET D=0
333 LET M=0
334 LET N=0
335 LET P=0
336 LET Q=0
337 LET R=0
338 LET S=0
339 LET H=0
340 LET D=0
341 LET M=0
342 LET N=0
343 LET P=0
344 LET Q=0
345 LET R=0
346 LET S=0
347 LET H=0
348 LET D=0
349 LET M=0
350 LET N=0
351 LET P=0
352 LET Q=0
353 LET R=0
354 LET S=0
355 LET H=0
356 LET D=0
357 LET M=0
358 LET N=0
359 LET P=0
360 LET Q=0
361 LET R=0
362 LET S=0
363 LET H=0
364 LET D=0
365 LET M=0
366 LET N=0
367 LET P=0
368 LET Q=0
369 LET R=0
370 LET S=0
371 LET H=0
372 LET D=0
373 LET M=0
374 LET N=0
375 LET P=0
376 LET Q=0
377 LET R=0
378 LET S=0
379 LET H=0
380 LET D=0
381 LET M=0
382 LET N=0
383 LET P=0
384 LET Q=0
385 LET R=0
386 LET S=0
387 LET H=0
388 LET D=0
389 LET M=0
390 LET N=0
391 LET P=0
392 LET Q=0
393 LET R=0
394 LET S=0
395 LET H=0
396 LET D=0
397 LET M=0
398 LET N=0
399 LET P=0
400 LET Q=0
401 LET R=0
402 LET S=0
403 LET H=0
404 LET D=0
405 LET M=0
406 LET N=0
407 LET P=0
408 LET Q=0
409 LET R=0
410 LET S=0
411 LET H=0
412 LET D=0
413 LET M=0
414 LET N=0
415 LET P=0
416 LET Q=0
417 LET R=0
418 LET S=0
419 LET H=0
420 LET D=0
421 LET M=0
422 LET N=0
423 LET P=0
424 LET Q=0
425 LET R=0
426 LET S=0
427 LET H=0
428 LET D=0
429 LET M=0
430 LET N=0
431 LET P=0
432 LET Q=0
433 LET R=0
434 LET S=0
435 LET H=0
436 LET D=0
437 LET M=0
438 LET N=0
439 LET P=0
440 LET Q=0
441 LET R=0
442 LET S=0
443 LET H=0
444 LET D=0
445 LET M=0
446 LET N=0
447 LET P=0
448 LET Q=0
449 LET R=0
450 LET S=0
451 LET H=0
452 LET D=0
453 LET M=0
454 LET N=0
455 LET P=0
456 LET Q=0
457 LET R=0
458 LET S=0
459 LET H=0
460 LET D=0
461 LET M=0
462 LET N=0
463 LET P=0
464 LET Q=0
465 LET R=0
466 LET S=0
467 LET H=0
468 LET D=0
469 LET M=0
470 LET N=0
471 LET P=0
472 LET Q=0
473 LET R=0
474 LET S=0
475 LET H=0
476 LET D=0
477 LET M=0
478 LET N=0
479 LET P=0
480 LET Q=0
481 LET R=0
482 LET S=0
483 LET H=0
484 LET D=0
485 LET M=0
486 LET N=0
487 LET P=0
488 LET Q=0
489 LET R=0
490 LET S=0
491 LET H=0
492 LET D=0
493 LET M=0
494 LET N=0
495 LET P=0
496 LET Q=0
497 LET R=0
498 LET S=0
499 LET H=0
500 LET D=0
501 LET M=0
502 LET N=0
503 LET P=0
504 LET Q=0
505 LET R=0
506 LET S=0
507 LET H=0
508 LET D=0
509 LET M=0
510 LET N=0
511 LET P=0
512 LET Q=0
513 LET R=0
514 LET S=0
515 LET H=0
516 LET D=0
517 LET M=0
518 LET N=0
519 LET P=0
520 LET Q=0
521 LET R=0
522 LET S=0
523 LET H=0
524 LET D=0
525 LET M=0
526 LET N=0
527 LET P=0
528 LET Q=0
529 LET R=0
530 LET S=0
531 LET H=0
532 LET D=0
533 LET M=0
534 LET N=0
535 LET P=0
536 LET Q=0
537 LET R=0
538 LET S=0
539 LET H=0
540 LET D=0
541 LET M=0
542 LET N=0
543 LET P=0
544 LET Q=0
545 LET R=0
546 LET S=0
547 LET H=0
548 LET D=0
549 LET M=0
550 LET N=0
551 LET P=0
552 LET Q=0
553 LET R=0
554 LET S=0
555 LET H=0
556 LET D=0
557 LET M=0
558 LET N=0
559 LET P=0
560 LET Q=0
561 LET R=0
562 LET S=0
563 LET H=0
564 LET D=0
565 LET M=0
566 LET N=0
567 LET P=0
568 LET Q=0
569 LET R=0
570 LET S=0
571 LET H=0
572 LET D=0
573 LET M=0
574 LET N=0
575 LET P=0
576 LET Q=0
577 LET R=0
578 LET S=0
579 LET H=0
580 LET D=0
581 LET M=0
582 LET N=0
583 LET P=0
584 LET Q=0
585 LET R=0
586 LET S=0
587 LET H=0
588 LET D=0
589 LET M=0
590 LET N=0
591 LET P=0
592 LET Q=0
593 LET R=0
594 LET S=0
595 LET H=0
596 LET D=0
597 LET M=0
598 LET N=0
599 LET P=0
600 LET Q=0
601 LET R=0
602 LET S=0
603 LET H=0
604 LET D=0
605 LET M=0
606 LET N=0
607 LET P=0
608 LET Q=0
609 LET R=0
610 LET S=0
611 LET H=0
612 LET D=0
613 LET M=0
614 LET N=0
615 LET P=0
616 LET Q=0
617 LET R=0
618 LET S=0
619 LET H=0
620 LET D=0
621 LET M=0
622 LET N=0
623 LET P=0
624 LET Q=0
625 LET R=0
626 LET S=0
627 LET H=0
628 LET D=0
629 LET M=0
630 LET N=0
631 LET P=0
632 LET Q=0
633 LET R=0
634 LET S=0
635 LET H=0
636 LET D=0
637 LET M=0
638 LET N=0
639 LET P=0
640 LET Q=0
641 LET R=0
642 LET S=0
643 LET H=0
644 LET D=0
645 LET M=0
646 LET N=0
647 LET P=0
648 LET Q=0
649 LET R=0
650 LET S=0
651 LET H=0
652 LET D=0
653 LET M=0
654 LET N=0
655 LET P=0
656 LET Q=0
657 LET R=0
658 LET S=0
659 LET H=0
660 LET D=0
661 LET M=0
662 LET N=0
663 LET P=0
664 LET Q=0
665 LET R=0
666 LET S=0
667 LET H=0
668 LET D=0
669 LET M=0
670 LET N=0
671 LET P=0
672 LET Q=0
673 LET R=0
674 LET S=0
675 LET H=0
676 LET D=0
677 LET M=0
678 LET N=0
679 LET P=0
680 LET Q=0
681 LET R=0
682 LET S=0
683 LET H=0
684 LET D=0
685 LET M=0
686 LET N=0
687 LET P=0
688 LET Q=0
689 LET R=0
690 LET S=0
691 LET H=0
692 LET D=0
693 LET M=0
694 LET N=0
695 LET P=0
696 LET Q=0
697 LET R=0
698 LET S=0
699 LET H=0
700 LET D=0
701 LET M=0
702 LET N=0
703 LET P=0
704 LET Q=0
705 LET R=0
706 LET S=0
707 LET H=0
708 LET D=0
709 LET M=0
710 LET N=0
711 LET P=0
712 LET Q=0
713 LET R=0
714 LET S=0
715 LET H=0
716 LET D=0
717 LET M=0
718 LET N=0
719 LET P=0
720 LET Q=0
721 LET R=0
722 LET S=0
723 LET H=0
724 LET D=0
725 LET M=0
726 LET N=0
727 LET P=0
728 LET Q=0
729 LET R=0
730 LET S=0
731 LET H=0
732 LET D=0
733 LET M=0
734 LET N=0
735 LET P=0
736 LET Q=0
737 LET R=0
738 LET S=0
739 LET H=0
740 LET D=0
741 LET M=0
742 LET N=0
743 LET P=0
744 LET Q=0
745 LET R=0
746 LET S=0
747 LET H=0
748 LET D=0
749 LET M=0
750 LET N=0
751 LET P=0
752 LET Q=0
753 LET R=0
754 LET S=0
755 LET H=0
756 LET D=0
757 LET M=0
758 LET N=0
759 LET P=0
760 LET Q=0
761 LET R=0
762 LET S=0
763 LET H=0
764 LET D=0
765 LET M=0
766 LET N=0
767 LET P=0
768 LET Q=0
769 LET R=0
770 LET S=0
771 LET H=0
772 LET D=0
773 LET M=0
774 LET N=0
775 LET P=0
776 LET Q=0
777 LET R=0
778 LET S=0
779 LET H=0
780 LET D=0
781 LET M=0
782 LET N=0
783 LET P=0
784 LET Q=0
785 LET R=0
786 LET S=0
787 LET H=0
788 LET D=0
789 LET M=0
790 LET N=0
791 LET P=0
792 LET Q=0
793 LET R=0
794 LET S=0
795 LET H=0
796 LET D=0
797 LET M=0
798 LET N=0
799 LET P=0
800 LET Q=0
801 LET R=0
802 LET S=0
803 LET H=0
804 LET D=0
805 LET M=0
806 LET N=0
807 LET P=0
808 LET Q=0
809 LET R=0
810 LET S=0
811 LET H=0
812 LET D=0
813 LET M=0
814 LET N=0
815 LET P=0
816 LET Q=0
817 LET R=0
818 LET S=0
819 LET H=0
820 LET D=0
821 LET M=0
822 LET N=0
823 LET P=0
824 LET Q=0
825 LET R=0
826 LET S=0
827 LET H=0
828 LET D=0
829 LET M=0
830 LET N=0
831 LET P=0
832 LET Q=0
833 LET R=0
834 LET S=0
835 LET H=0
836 LET D=0
837 LET M=0
838 LET N=0
839 LET P=0
840 LET Q=0
841 LET R=0
842 LET S=0
843 LET H=0
844 LET D=0
845 LET M=0
846 LET N=0
847 LET P=0
848 LET Q=0
849 LET R=0
850 LET S=0
851 LET H=0
852 LET D=0
853 LET M=0
854 LET N=0
855 LET P=0
856 LET Q=0
857 LET R=0
858 LET S=0
859 LET H=0
860 LET D=0
861 LET M=0
862 LET N=0
863 LET P=0
864 LET Q=0
865 LET R=0
866 LET S=0
867 LET H=0
868 LET D=0
869 LET M=0
870 LET N=0
871 LET P=0
872 LET Q=0
873 LET R=0
874 LET S=0
875 LET H=0
876 LET D=0
877 LET M=0
878 LET N=0
879 LET P=0
880 LET Q=0
881 LET R=0
882 LET S=0
883 LET H=0
884 LET D=0
885 LET M=0
886 LET N=0
887 LET P=0
888 LET Q=0
889 LET R=0
890 LET S=0
891 LET H=0
892 LET D=0
893 LET M=0
894 LET N=0
895 LET P=0
896 LET Q=0
897 LET R=0
898 LET S=0
899 LET H=0
900 LET D=0
901 LET M=0
902 LET N=0
903 LET P=0
904 LET Q=0
905 LET R=0
906 LET S=0
907 LET H=0
908 LET D=0
909 LET M=0
910 LET N=0
911 LET P=0
912 LET Q=0
913 LET R=0
914 LET S=0
915 LET H=0
916 LET D=0
917 LET M=0
918 LET N=0
919 LET P=0
920 LET Q=0
921 LET R=0
922 LET S=0
923 LET H=0
924 LET D=0
925 LET M=0
926 LET N=0
927 LET P=0
928 LET Q=0
929 LET R=0
930 LET S=0
931 LET H=0
932 LET D=0
933 LET M=0
934 LET N=0
935 LET P=0
936 LET Q=0
937 LET R=0
938 LET S=0
939 LET H=0
940 LET D=0
941 LET M=0
942 LET N=0
943 LET P=0
944 LET Q=0
945 LET R=0
946 LET S=0
947 LET H=0
948 LET D=0
949 LET M=0
950 LET N=0
951 LET P=0
952 LET Q=0
953 LET R=0
954 LET S=0
955 LET H=0
956 LET D=0
957 LET M=0
958 LET N=0
959 LET P=0
960 LET Q=0
961 LET R=0
962 LET S=0
963 LET H=0
964 LET D=0
965 LET M=0
966 LET N=0
967 LET P=0
968 LET Q=0
969 LET R=0
970 LET S=0
971 LET H=0
972 LET D=0
973 LET M=0
974 LET N=0
975 LET P=0
976 LET Q=0
977 LET R=0
978 LET S=0
979 LET H=0
980 LET D=0
981 LET M=0
982 LET N=0
983 LET P=0
984 LET Q=0
985 LET R=0
986 LET S=0
987 LET H=0
988 LET D=0
989 LET M=0
990 LET N=0
991 LET P=0
992 LET Q=0
993 LET R=0
994 LET S=0
995 LET H=0
996 LET D=0
997 LET M=0
998 LET N=0
999 LET P=0
1000 LET Q=0
1001 LET R=0
1002 LET S=0
1003 LET H=0
1004 LET D=0
1005 LET M=0
1006 LET N=0
1007 LET P=0
1008 LET Q=0
1009 LET R=0
1010 LET S=0
1011 LET H=0
1012 LET D=0
1013 LET M=0
1014 LET N=0
1015 LET P=0
1016 LET Q=0
1017 LET R=0
1018 LET S=0
1019 LET H=0
1020 LET D=0
1021 LET M=0
1022 LET N=0
1023 LET P=0
1024 LET Q=0
1025 LET R=0
1026 LET S=0
1027 LET H=0
1028 LET D=0
1029 LET M=0
1030 LET N=0
1031 LET P=0
1032 LET Q=0
1033 LET R=0
1034 LET S=0
1035 LET H=0
1036 LET D=0
1037 LET M=0
1038 LET N=0
1039 LET P=0
1040 LET Q=0
1041 LET R=0
1042 LET S=0
1043 LET H=0
1044 LET D=0
1045 LET M=0
1046 LET N=0
1047 LET P=0
1048 LET Q=0
1049 LET R=0
1050 LET S=0
1051 LET H=0
1052 LET D=0
1053 LET M=0
1054 LET N=0
1055 LET P=0
1056 LET Q=0
1057 LET R=0
1058 LET S=0
1059 LET H=0
1060 LET D=0
1061 LET M=0
1062 LET N=0
1063 LET P=0
1064 LET Q=0
1065 LET R=0
1066 LET S=0
1067 LET H=0
1068 LET D=0
1069 LET M=0
1070 LET N=0
1071 LET P=0
1072 LET Q=0
1073 LET R=0
1074 LET S=0
1075 LET H=0
1076 LET D=0
1077 LET M=0
1078 LET N=0
1079 LET P=0
1080 LET Q=0
1081 LET R=0
1082 LET S=0
1083 LET H=0
1084 LET D=0
1085 LET M=0
1086 LET N=0
1087 LET P=0
1088 LET Q=0
1089 LET R=0
1090 LET S=0
1091 LET H=0
1092 LET D=0
1093 LET M=0
1094 LET N=0
1095 LET P=0
1096 LET Q=0
1097 LET R=0
1098 LET S=0
1099 LET H=0
1100 LET D=0
1101 LET M=0
1102 LET N=0
1103 LET P=0
1104 LET Q=0
1105 LET R=0
1106 LET S=0
1107 LET H=0
1108 LET D=0
1109 LET M=0
1110 LET N=0
1111 LET P=0
1112 LET Q=0
1113 LET R=0
1114 LET S=0
1115 LET H=0
1116 LET D=0
1117 LET M=0
1118 LET N=0
1119 LET P=0
1120 LET Q=0
1121 LET R=0
1122 LET S=0
1123 LET H=0
1124 LET D=0
1125 LET M=0
1126 LET N=0
1127 LET P=0
1128 LET Q=0
1129 LET R=0
1130 LET S=0
1131 LET H=0
1132 LET D=0
1133 LET M=0
1134 LET N=0
1135 LET P=0
1136 LET Q=0
1137 LET R=0
1138 LET S=0
1139 LET H=0
1140 LET D=0
1141 LET M=0
1142 LET N=0
1143 LET P=0
1144 LET Q=0
1145 LET R=0
1146 LET S=0
1147 LET H=0
1148 LET D=0
1149 LET M=0
1150 LET N=0
1151 LET P=0
1152 LET Q=0
1153 LET R=0
1154 LET S=0
1155 LET H=0
1156 LET D=0
1157 LET M=0
1158 LET N=0
1159 LET P=0
1160 LET Q=0
1161 LET R=0
1162 LET S=0
1163 LET H=0
1164 LET D=0
1165 LET M=0
1166 LET N=0
1167 LET P=0
1168 LET Q=0
1169 LET R=0
1170 LET S=0
1171 LET H=0
1172 LET D=0
1173 LET M=0
1174 LET N=0
1175 LET P=0
1176 LET Q=0
1177 LET R=0
1178 LET S=0
1179 LET H=0
1180 LET D=0
1181 LET M=0
1182 LET N=0
1183 LET P=0
1184 LET Q=0
1185 LET R=0
1186 LET S=0
1187 LET H=0
1188 LET D=0
1189 LET M=0
1190 LET N=0
1191 LET P=0
1192 LET Q=0
1193 LET R=0
1194 LET S=0
1195 LET H=0
1196 LET D=0
1197 LET M=0
1198 LET N=0
1199 LET P=0
1200 LET Q=0
1201 LET R=0
1202 LET S=0
1203 LET H=0
1204 LET D=0
1205 LET M=0
1206 LET N=0
1207 LET P=0
1208 LET Q=0
1209 LET R=0
1210 LET S=0
1211 LET H=0
1212 LET D=0
1213 LET M=0
1214 LET N=0
1215 LET P=0
1216 LET Q=0
1217 LET R=0
1218 LET S=0
1219 LET H=0
1220 LET D=0
1221 LET M=0
1222 LET N=0
1223 LET P=0
1224 LET Q=0
1225 LET R=0
1226 LET S=0
1227 LET H=0
1228 LET D=0
1229 LET M=0
1230 LET N=0
1231 LET P=0
1232 LET Q=0
1233 LET R=0
1234 LET S=0
1235 LET H=0
1236 LET D=0
1237 LET M=0
1238 LET N=0
1239 LET P=0
1240 LET Q=0
1241 LET R=0
1242 LET S=0
1243 LET H=0
1244 LET D=0
1245 LET M=0
1246 LET N=0
1247 LET P=0
1248 LET Q=0
1249 LET R=0
1250 LET S=0
1251 LET H=0
1252 LET D=0
1253 LET M=0
1254 LET N=0
1255 LET P=0
1256 LET Q=0
1257 LET R=0
1258 LET S=0
1259 LET H=0
1260 LET D=0
1261 LET M=0
1262 LET N=0
1263 LET P=0
1264 LET Q=0
1265 LET R=0
1266 LET S=0
1267 LET H=0
1268 LET D=0
1269 LET M=0
1270 LET N=0
1271 LET P=0
1272 LET Q=0
1273 LET R=0
1274 LET S=0
1275 LET H=0
1276 LET D=0
1277 LET M=0
1278 LET N=0
1279 LET P=0
1280 LET Q=0
1281 LET R=0
1282 LET S=0
1283 LET H=0
1284 LET D=0
1285 LET M=0
1286 LET N=0
1287 LET P=0
1288 LET Q=0
1289 LET R=0
1290 LET S=0
1291 LET H=0
1292 LET D=0
1293 LET M=0
1294 LET N=0
1295 LET P=0
1296 LET Q=0
1297 LET R=0
1298 LET S=0
1299 LET H=0
1300 LET D=0
1301 LET M=0
1302 LET N=0
1303 LET P=0
1304 LET Q=0
1305 LET R=0
1306 LET S=0
1307 LET H=0
1308 LET D=0
1309 LET M=0
1310 LET N=0
1311 LET P=0
1312 LET Q=0
1313 LET R=0
1314 LET S=0
1315 LET H=0
1316 LET D=0
1317 LET M=0
1318 LET N=0
1319 LET P=0
1320 LET Q=0
1321 LET R=0
1322 LET S=0
1323 LET H=0
1324 LET D=0
1325 LET M=0
1326 LET N=0
1327 LET P=0
1328 LET Q=0
1329 LET R=0
1330 LET S=0
1331 LET H=0
1332 LET D=0
1333 LET M=0
1334 LET N=0
1335 LET P=0
1336 LET Q=0
1337 LET R=0
1338 LET S=0
1339 LET H=0
1340 LET D=0
1341 LET M=0
1342 LET N=0
1343 LET P=0
1344 LET Q=0
1345 LET R=0
1346 LET S=0
1347 LET H=0
1348 LET D=0
1349 LET M=0
1350 LET N=0
1351 LET P=0
1352 LET Q=0
1353 LET R=0
1354 LET S=0
1355 LET H=0
1356 LET D=0
1357 LET M=0
1358 LET N=0
1359 LET P=0
1360 LET Q=0
1361 LET R=0
1362 LET S=0
1363 LET H=0
1364 LET D=0
1365 LET M=0
1366 LET N=0
1367 LET P=0
1368 LET Q=0
1369 LET R=0
1370 LET S=0
1371 LET H=0
1372 LET D=0
1373 LET M=0
1374 LET N=0
1375 LET P=0
1376 LET Q=0
1377 LET R=0
1378 LET S=0
1379 LET H=0
1380 LET D=0
1381 LET M=0
1382 LET N=0
1383 LET P=0
1384 LET Q=0
1385 LET R=0
1386 LET S=0
1387 LET H=0
1388 LET D=0
1389 LET M=0
1390 LET N=0
1391 LET P=0
1392 LET Q=0
1393 LET R=0
1394 LET S=0
1395 LET H=0
1396
```



```

0001 IF X=25 THEN PRINT AT U,M-1 1232 LET H=0+1
0002 IF X=25 THEN LET H=M-1 1233 GOTO 1255
0003 IF X=25 THEN PRINT AT U,M-1 1234 RETURN
0004 1235 CLS
0005 PRINT AT 8,9:"CRAZY BALLOON"
0006 PRINT AT 1,5:"-----"
0007 PRINT "MOVE AROUND THE COURSE"
0008 PRINT "E AVOIDING:"
0009 PRINT "THE PRICKLY STARS:-"
0010 PRINT
0011 PRINT
0012 PRINT "ENTER POINTS ARE ALL"
0013 PRINT "GOING THROUGH THE GOAL"
0014 PRINT
0015 PRINT "THE OBJECT IS TO GET"
0016 PRINT "GOAL IN THE TOP LEFT"
0017 PRINT "CORNER."
0018 PRINT "THE CONTROLS ARE:"
0019 PRINT
0020 PRINT AT 10,10:"GO"
0021 PRINT AT 10,12:"LEFT"
0022 PRINT AT 10,15:"RIGHT"
0023 CLS
0024 GOTO 25
0025 LET V=U-1
0026 LET H=M-1
0027 LET Y=Y-1
0028 LET Z=Z-1
0029 PRINT AT V+1,H-1," "
0030 PRINT AT V+1,Z-1," "
0031 GOTO 125
0032 LET U=U-1
0033 LET H=M-1
0034 LET Y=Y-1
0035 LET Z=Z-1
0036 PRINT AT V+1,H+1," "
0037 PRINT AT V+1,Z+1," "
0038 GOTO 125
0039 LET U=U+1
0040 LET H=M+1
0041 LET Y=Y+1
0042 LET Z=Z+1
0043 PRINT AT V-1,H-1," "
0044 PRINT AT V-1,Z-1," "
0045 GOTO 125
0046 LET U=U+1
0047 LET H=M+1
0048 LET Y=Y+1
0049 LET Z=Z+1
0050 PRINT AT V-1,H+1," "
0051 PRINT AT V-1,Z+1," "
0052 GOTO 125
0053 LET H=M-1
0054 LET Z=Z-1
0055 PRINT AT V,H+1," "
0056 PRINT AT V,Z+1," "
0057 GOTO 125
0058 LET H=M+1
0059 LET Z=Z+1
0060 PRINT AT V,H-1," "
0061 PRINT AT V,Z-1," "
0062 GOTO 125
0063 LET U=U-1
0064 LET Y=Y-1
0065 PRINT AT V+1,H," "
0066 PRINT AT V+1,Z," "
0067 GOTO 125
0068 LET V=V-1
0069 LET Y=Y-1
0070 PRINT AT V+1,H," "
0071 PRINT AT V+1,Z," "
0072 GOTO 125
0073 LET V=V+1
0074 LET Y=Y+1
0075 PRINT AT V-1,H," "
0076 PRINT AT V-1,Z," "
0077 GOTO 125
0078 LET V=V+1
0079 LET Y=Y+1
0080 PRINT AT V-1,H," "
0081 PRINT AT V-1,Z," "
0082 GOTO 125
0083 LET V=V-1
0084 LET Y=Y-1
0085 PRINT AT V+1,H," "
0086 PRINT AT V+1,Z," "
0087 GOTO 125
0088 LET V=V-1
0089 LET Y=Y-1
0090 PRINT AT V+1,H," "
0091 PRINT AT V+1,Z," "
0092 GOTO 125
0093 LET V=V+1
0094 LET Y=Y+1
0095 PRINT AT V-1,H," "
0096 PRINT AT V-1,Z," "
0097 GOTO 125
0098 LET V=V+1
0099 LET Y=Y+1
0100 PRINT AT V-1,H," "
0101 PRINT AT V-1,Z," "
0102 GOTO 125
0103 LET V=V-1
0104 LET Y=Y-1
0105 PRINT AT V+1,H," "
0106 PRINT AT V+1,Z," "
0107 GOTO 125
0108 LET V=V-1
0109 LET Y=Y-1
0110 PRINT AT V+1,H," "
0111 PRINT AT V+1,Z," "
0112 GOTO 125
0113 LET V=V+1
0114 LET Y=Y+1
0115 PRINT AT V-1,H," "
0116 PRINT AT V-1,Z," "
0117 GOTO 125
0118 LET V=V+1
0119 LET Y=Y+1
0120 PRINT AT V-1,H," "
0121 PRINT AT V-1,Z," "
0122 GOTO 125
0123 IF H=1 THEN GOTO 1255

```

This is the building for the program, *Green Meadows*.

WORMS



This is a two player game, the object being to force your opponent to collide with a wall, your worm or your opponent's worm.

The game uses the full 24 by 32 screen, the walls being the black border around the screen. One worm is made from '+' characters, the other is made from 'v' characters. The worms grow as you move around the screen.

Movement is in four directions: up, down, left and right. Keys 'W' (up), 'A' (left), 'S' (right) and 'Z' (down) control the '+' worm. Keys 'O' (up), 'K' (left), 'L' (right) and ';' (down) control the 'v' worm. A worm is killed if it hits a wall, hits the opponent's worm, collides with itself or moves back on itself. The game also has on-screen scoring for both players. Key 'P' returns control to BASIC and stops the program.

DOWN THE LINE

Looking at the main program, the machine code is held in the line 1 REM statement containing 532 characters. There is an easy way and a hard way to create such a REM statement. The hard way is to type in the 532 characters one after another. The easy way is as follows: create the REM statement in Fig. 1, that is, a REM followed by 100 characters. Edit the line numbers to form lines 2 to 5 and then enter line 6 (a REM followed by two characters) as

in Fig. 2. Next, execute the following as direct commands:

```
POKE 16910,0 (this ensures the
line cannot be edited)
POKE 16911,32
POKE 16912,2
```

A REM containing 532 characters has now been created, and the machine code program can now be entered using a line loader like that in Fig. 3. Incidentally, if PRINT PEEK (16911) + 256*PEEK (16912) does not give the answer 532 then you have typed in the REMs wrongly, and you should go through the above procedure again.

After you have entered the machine code program, you can check to see if you have entered it correctly, using the program in Fig. 4. When you are sure you have entered the machine code program correctly, you can now enter the BASIC control program. Save the program first by RUN 200, just in case something goes wrong.

TOO FAST?

To increase the speed of the game, type in the following as direct commands:

```
POKE 16912,X (Where X is a
value between 200 and 255.
The lower the value, the faster
the game.)
```

If you can worm your way round the machine code, you'll find it's worth it.

POKE 16913,0

To slow the game down, type in the following:

```
POKE 16912,X (Where X has
between zero and 255. The
greater the value the slower the
game.)
```

```
POKE 16913,Y (Y or 'Z' is
usually a big enough value for
Y).
```

If you would like to change the characters which form the worms, then change the '+' by typing in the following:

```
POKE 16898,C
POKE 16797,C+128 (inverse of
C.)
POKE 16870,C
POKE 16800,C
POKE 16905,C
```

where C is the code of the required character.

To change the 'v' worm, type in the following:

```
POKE 16890,C
POKE 16875,C
POKE 16900,C
POKE 16910,C
POKE 16913,C+128
```

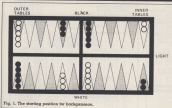
The program, as published, was originally written for a 16K ZX80, although with the addition of a little bit of machine code it will run on a machine which has 3K of RAM, ie where the display file is not automatically filled with spaces. To make the program run in 3K, the machine code should be placed in a line 2 REM statement.

```
LD HL,4032      31 23 40
LD (HL),00      30 00
LD A,00          3E 00
LD C,18          0E 18
LD B,20 NEXTLN  00 20
RST 16 NEXTCHR  D7
CONV NEXTCHR    10 FD
DEC C           00
JR NZ,NEXTLN    30 FB
RET             C9
```


Address	Instruction	Register	Value	Register	Value
0000	CALL MOVE			COSE41	
0001	LD 148887, HL			252548	
0002	LD R, HL			FE00	
0003	LD R, HL			FE00	
0004	LD R, HL			FE00	
0005	LD R, HL			FE00	
0006	LD R, HL			FE00	
0007	LD R, HL			FE00	
0008	LD R, HL			FE00	
0009	LD R, HL			FE00	
000A	LD R, HL			FE00	
000B	LD R, HL			FE00	
000C	LD R, HL			FE00	
000D	LD R, HL			FE00	
000E	LD R, HL			FE00	
000F	LD R, HL			FE00	
0010	LD R, HL			FE00	
0011	LD R, HL			FE00	
0012	LD R, HL			FE00	
0013	LD R, HL			FE00	
0014	LD R, HL			FE00	
0015	LD R, HL			FE00	
0016	LD R, HL			FE00	
0017	LD R, HL			FE00	
0018	LD R, HL			FE00	
0019	LD R, HL			FE00	
001A	LD R, HL			FE00	
001B	LD R, HL			FE00	
001C	LD R, HL			FE00	
001D	LD R, HL			FE00	
001E	LD R, HL			FE00	
001F	LD R, HL			FE00	
0020	LD R, HL			FE00	
0021	LD R, HL			FE00	
0022	LD R, HL			FE00	
0023	LD R, HL			FE00	
0024	LD R, HL			FE00	
0025	LD R, HL			FE00	
0026	LD R, HL			FE00	
0027	LD R, HL			FE00	
0028	LD R, HL			FE00	
0029	LD R, HL			FE00	
002A	LD R, HL			FE00	
002B	LD R, HL			FE00	
002C	LD R, HL			FE00	
002D	LD R, HL			FE00	
002E	LD R, HL			FE00	
002F	LD R, HL			FE00	
0030	LD R, HL			FE00	
0031	LD R, HL			FE00	
0032	LD R, HL			FE00	
0033	LD R, HL			FE00	
0034	LD R, HL			FE00	
0035	LD R, HL			FE00	
0036	LD R, HL			FE00	
0037	LD R, HL			FE00	
0038	LD R, HL			FE00	
0039	LD R, HL			FE00	
003A	LD R, HL			FE00	
003B	LD R, HL			FE00	
003C	LD R, HL			FE00	
003D	LD R, HL			FE00	
003E	LD R, HL			FE00	
003F	LD R, HL			FE00	
0040	LD R, HL			FE00	
0041	LD R, HL			FE00	
0042	LD R, HL			FE00	
0043	LD R, HL			FE00	
0044	LD R, HL			FE00	
0045	LD R, HL			FE00	
0046	LD R, HL			FE00	
0047	LD R, HL			FE00	
0048	LD R, HL			FE00	
0049	LD R, HL			FE00	
004A	LD R, HL			FE00	
004B	LD R, HL			FE00	
004C	LD R, HL			FE00	
004D	LD R, HL			FE00	
004E	LD R, HL			FE00	
004F	LD R, HL			FE00	
0050	LD R, HL			FE00	
0051	LD R, HL			FE00	
0052	LD R, HL			FE00	
0053	LD R, HL			FE00	
0054	LD R, HL			FE00	
0055	LD R, HL			FE00	
0056	LD R, HL			FE00	
0057	LD R, HL			FE00	
0058	LD R, HL			FE00	
0059	LD R, HL			FE00	
005A	LD R, HL			FE00	
005B	LD R, HL			FE00	
005C	LD R, HL			FE00	
005D	LD R, HL				

BACKGAMMON

Here's a classic game brought up to date with a program for the ZX81 that plays against you and obeys all the rules.



Backgammon is one of the more popular and enduring of board games, probably because it doesn't rely solely on blind chance and the throw of the dice, but requires an appreciation of probabilities and the skill to assess the most

effective use of your throws. This program puts a simulation of the backgammon board on your TV screen, and allows you to play against the computer. We aren't pretending that this piece of software could wipe the floor with Oscar Sharp, but it will put up a creditable fight.

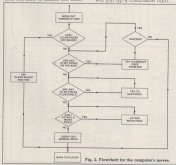
and can usually be guaranteed to cause you the maximum difficulty if the dice are with it. The performance is especially good when you realize that there isn't enough memory in a 16k Z801 for a look-ahead probability tree even one turn deep.

Just to make sure that you don't get tempted into cheating it, the computer checks out everything you enter against the rules of backgammon and rejects invalid moves. The one exception is the rule stating that if a player can only use one of the numbers he has rolled, he must, if possible, use the larger. This was in the copy of rules we obtained, but two backgammon players in the office said they'd never heard of it, so we decided to leave it out. If you play by this rule, it wouldn't require more than about five or six extra lines of code; this is left as an exercise to the reader!

SIMULATION

The working of the program is based on the two position arrays W[] and B[], which correspond to the points on the backgammon board. The board consists of 24 positions, or points, each of which may hold any number of pieces (of the same colour) up to the total number in play. Two array elements B[0]-B[24] hold the number of black pieces per position, while the corresponding information for the white player is held in W[0]-W[24]. For example, at the start of the game B[1] is set to two, since there are two black pieces at position A on the board. During the course of the game these array elements are constantly changing.

Two further elements of each array are used in the game, these being W33H, H33H, W33S and B33S. The



first two hold the number of pieces each player has on the bar, while the second two hold the number of pieces each player has taken off the board at the end of the game.

The computer uses the position arrays to work out its move. For every move that the computer makes, the routine shown in the flowchart of Figure 3 will be carried out. To give an example, suppose one of the computer's throws is a five. The program will look at each black piece, take the index of its position in the array and add five to it. It can then see the effect of moving the piece five places — whether the move is legal, whether a white piece could be taken, whether a position could be fortified and so on. If the computer is satisfied that the move is a good one, it is carried out.

Things get more complicated when the computer has to check whether a move that you have made is legal. The player's dice throws are stored in a third array, T1(), to simplify the checking routine by using FOR-NEXT loops. If the dice throws are not equal, the two numbers are put into T1() and T2() in descending order (using TC9) as a temporary store and the remaining array elements made zero. Otherwise doubles have been thrown, and the first four elements of the array are all made equal. TC9 again ending up as zero.

If a player's move is judged to be valid, the dice throw corresponding to that move must be tagged as 'used' by setting it to an 'impossible' value. We decided that the answer was to use 42.

Since the three arrays completely describe the state of the game with respect to piece positions and dice in play, the computer can run through a series of tests involving AND, OR and the arithmetic operators, one test per rule. As an example, the test that a player is genuinely unable to move is made between lines 1300 and 1450. For each unused throw of the dice and each point on the board, line 1400 checks that the movement of



'destination' point for White really is blocked by Black. Then line 1400 checks that, if White is on the bar, he generously cannot move a piece back into play.

An added complication is introduced when White starts to hear off, since it is possible to use a dice throw greater than the distance the rook is moved.

Additional logic between lines 680 and 770 takes care of the

PLAY THE GAME

It is assumed that the user of this program already knows how to play backgammon — it is not a teaching aid. The following instructions are essential to those

VARIABLE	FUNCTION
C (- M)	Random number used for computer's first throw
D (- V)	Random number used for computer's second throw
G\$,GG	Player's move (from)
H\$,HH	Player's move (to)
DD	Distance of a player's move
P,I,K,L N,S,C,T U,X,CC,TT	General purpose loop counters and flags
WT)	Position array for white pieces
BT)	Position array for black pieces
W(31)	White pieces out of play on the bar
B(31)	Black pieces out of play on the bar
W(36)	White pieces moved off the board
B(36)	Black pieces moved off the board
T)	Array containing White's dice throws

PERSONNEL STRUCTURE

10-190	Dimensions and initialize arrays. Initialize variables.
200	Generate a random number and decide who goes first.
220-330	Throw random numbers for player's dice and display them. Sort them into descending order for testing later. Check for doubles and award bonus counter and array if necessary.

Importing player's records

370-540 Display appropriate prompts and accept
input. Convert strings to character codes,
then subtract offset to put them in the range
1-24. Make preliminary validity checks, and
alter GS or HH to appropriate values if
"BAH" or "BEST" have been input.
520-630 If White is bearing off, check that move is

how to play the game with the 2888.

The player always plays White and should move in an anti-clockwise direction from I to A (see the photograph). When the program is first run, the computer may move first: if this has happened, the computer's dice throws and moves are displayed at the top of the screen. To throw your own dice, press any key, and after eight seconds or so the computer will display the board and your throws. This will happen anyway if the computer has allowed you to move first. (If you would prefer the computer to automatically throw your dice for you after an eight-second pause, delete lines 3190-3210 and change line 3300 to

3230 PAUSE 400

Pressing any key will terminate the pause early, and you can change the delay to suit your needs).

The player will then be asked for his move with the prompt "YOUR MOVE FROM" — you should enter the letter of the point you want to move from, then press NEWLINE. The prompt "TO" then appears, and the destination point is entered in a similar way. If the move is invalid, the computer will say so and repeat the above procedure.

If you can't move you should reply "NO" to the prompt "YOUR MOVE FROM" and the computer will make its next set of moves. If you've made a mistake in entering the "MOVE FROM" letter, you can cancel the move by typing "NO" in response to the "TO" prompt. The computer will then go back and ask for the move again. Once the computer has accepted a move as valid, you cannot change it.

When moving one of your pieces back into play from the bar, enter "BAR" in response to the first prompt. If you are moving one of your pieces off the board at the end of the game, enter "REST" in response to the prompt "TO".

If you throw a double, you are required to make four sets of moves.

valid because no pieces are on the bar or off the inner table.

Set move distance variable.

Test that move distance corresponds to one of the dice throws. If White isn't bearing off, jump directly to 840 to test flag.

Check that, if player is bearing off with a dice throw greater than required, no White pieces are on higher point. Check that a dice throw starts to permit the required move, and set DD to this value ready for the next routine. Check that a dice throw exists to permit the required move, and change the array values to register White's move if it's valid. Check for a win by White and print appropriate messages. Stop program.

Check if any of Black's pieces have been taken and alter array if necessary. Loop for next move.

Go to Black's move.

Routine to display invalid move.

Computer's move

Subroutine to check whether the computer can start moving its pieces off the board (similar to routine for White above).

Computer decides whether to make an ordinary move or move pieces off the board. Remove piece from board with first move. Remove piece from board with second move. Display the move and change the array. Check if Black has won and display appropriate messages.

Entry point to computer's move routine if White has been unable to move. This is checked and a jump made to the invalid message routine if necessary. Otherwise fall through.

Computer's throws calculated and displayed. Loop counter altered if doubles thrown. Check to see whether any of computer's pieces are on the bar. If so, go to the bar-clearing routine at line 3120.

Call subroutine to check if computer can bear off. Go to bearing-off routine at 1080 if so. Decide whether to play defensively or not (go to line 1600 if not).

Check if any pieces are vulnerable. Go to routine at 2010 if any are.

First move (attack white pieces. Check to see if a black piece can move from the point in question. If a white piece is available for taking on that spot, go to the 'take piece' routine at 2870. An identical routine for the second move is at lines 1870-1900.

First move (if no white pieces can be attacked). Check if a black piece can be moved legally and do so. An identical routine for the second move is at lines 1940-2000.

Routine for the saving of exposed pieces. Check to see if a white piece can be taken with either the first move (3000), or the second move (2050), or with both moves added together (3000). If a white piece can be taken in any of these ways, the appropriate action is taken. If no white piece is exposed,

a check is carried out to see if an ordinary move is possible for the piece (2100 and 2130). If one is found it is implemented.

Drawing the board

- 2160-2240 Plot in frame.
2250-2360 Print top and bottom of frame using inverse video characters.
2370-2600 Display pieces on board using PRINT AT. If more than five pieces are on a point, replace top piece with number of pieces.
2610-2740 If either player has pieces on the bar, display them at the side of the board.

Registering Black's Moves

- 2750-2860 Registering of moves if no white piece has been taken (2750-2800 for the first move, 2810-2860 for the second). Moves displayed on screen by lines 2780 and 2840.
2870-3020 Registering of move if white piece has been taken (2870-2940 for the first move, 2950-3020 for the second). Moves displayed on screen by lines 2920 and 3000. These moves differ from the previous set because W(3) is incremented by one and the array position that the white piece occupied is decremented by one. The player is informed that a piece has been taken by the message "BAR".
3030-3110 This routine is used if the computer has used both its throws together to take a white piece. If the computer has a piece or pieces on the bar, this routine decides the printout format to be used. Line 3050 is chosen for the first move, line 3070 for the second. If the computer is unable to move, line 3150 ensures that the appropriate message is printed once only.
3160 If the computer has thrown a double, this line sends the computer back to the beginning of the movement routine to make its second two moves.
3170 Displays the board with the new moves on it.
3180-3240 Waits for the player to press a key before continuing with the next throw of the dice. May be altered to a PAUSE statement for automatic play.
3250-3480 Register moves if the computer has brought a piece back into play from the bar (3250-3360 for the first move, 3370-3480 for the second). If a white piece has been taken too, the routines at 3390 and 3430 are called.

Should either you or the computer get more than five pieces on any point, a complication arises because of the restricted height of the ZX81 display. We got round this by replacing the top piece on the point with the number of pieces on the point (like photographs make this clear).

CONVERSIONS

Conversion to other, more sophisticated machines should prove no problem at all. The initial board position could be set up by using READ-DATA statements. CLEAR (line 30) may need a figure after it on some machines. If you use the PAUSE version, the delay could be obtained with a FOR-NEXT loop. RAND need not be used on some machines, but may be found in the form RANDOMIZE on others. CODE is normally found in the form ASCII, and a further complication is introduced by the non-standard ZX81 character set. On a machine using standard ASCII codes, the offset 37 in lines 390, 460, 1190, 2280, 2840, 2920, 3000, 3080, 3270, and 3390 would have to be changed to 64. END is usually END\$ or END!).

As far as the screen display goes, much better simulations of the board should be possible on machines like the Dragon 32, BBC and ZX Spectrum due to the higher resolution and colour. If you have a machine which doesn't support PRINT AT or a similar statement, you could adapt the system we used in The Valley with HOME1 and cursor movement strings.

```

100 DIM C(12,12)
110 LET U=0
120 READ R1,R2
130 LET C(0,0)=R1
140 LET C(0,1)=R2
150 LET C(1,0)=R1
160 LET C(1,1)=R2
170 LET C(2,0)=R1
180 LET C(2,1)=R2
190 LET C(3,0)=R1
200 LET C(3,1)=R2
210 LET C(4,0)=R1
220 LET C(4,1)=R2
230 LET C(5,0)=R1
240 LET C(5,1)=R2
250 LET C(6,0)=R1
260 LET C(6,1)=R2
270 LET C(7,0)=R1
280 LET C(7,1)=R2
290 LET C(8,0)=R1
300 LET C(8,1)=R2
310 LET C(9,0)=R1
320 LET C(9,1)=R2
330 LET C(10,0)=R1
340 LET C(10,1)=R2
350 LET C(11,0)=R1
360 LET C(11,1)=R2
370 LET C(12,0)=R1
380 LET C(12,1)=R2
390 LET C(13,0)=R1
400 LET C(13,1)=R2
410 LET C(14,0)=R1
420 LET C(14,1)=R2
430 LET C(15,0)=R1
440 LET C(15,1)=R2
450 LET C(16,0)=R1
460 LET C(16,1)=R2
470 LET C(17,0)=R1
480 LET C(17,1)=R2
490 LET C(18,0)=R1
500 LET C(18,1)=R2
510 LET C(19,0)=R1
520 LET C(19,1)=R2
530 LET C(20,0)=R1
540 LET C(20,1)=R2
550 LET C(21,0)=R1
560 LET C(21,1)=R2
570 LET C(22,0)=R1
580 LET C(22,1)=R2
590 LET C(23,0)=R1
600 LET C(23,1)=R2
610 LET C(24,0)=R1
620 LET C(24,1)=R2
630 LET C(25,0)=R1
640 LET C(25,1)=R2
650 LET C(26,0)=R1
660 LET C(26,1)=R2
670 LET C(27,0)=R1
680 LET C(27,1)=R2
690 LET C(28,0)=R1
700 LET C(28,1)=R2
710 LET C(29,0)=R1
720 LET C(29,1)=R2
730 LET C(30,0)=R1
740 LET C(30,1)=R2
750 LET C(31,0)=R1
760 LET C(31,1)=R2
770 LET C(32,0)=R1
780 LET C(32,1)=R2
790 LET C(33,0)=R1
800 LET C(33,1)=R2
810 LET C(34,0)=R1
820 LET C(34,1)=R2
830 LET C(35,0)=R1
840 LET C(35,1)=R2
850 LET C(36,0)=R1
860 LET C(36,1)=R2
870 LET C(37,0)=R1
880 LET C(37,1)=R2
890 LET C(38,0)=R1
900 LET C(38,1)=R2
910 LET C(39,0)=R1
920 LET C(39,1)=R2
930 LET C(40,0)=R1
940 LET C(40,1)=R2
950 LET C(41,0)=R1
960 LET C(41,1)=R2
970 LET C(42,0)=R1
980 LET C(42,1)=R2
990 LET C(43,0)=R1
1000 LET C(43,1)=R2
1010 LET C(44,0)=R1
1020 LET C(44,1)=R2
1030 LET C(45,0)=R1
1040 LET C(45,1)=R2
1050 LET C(46,0)=R1
1060 LET C(46,1)=R2
1070 LET C(47,0)=R1
1080 LET C(47,1)=R2
1090 LET C(48,0)=R1
1100 LET C(48,1)=R2
1110 LET C(49,0)=R1
1120 LET C(49,1)=R2
1130 LET C(50,0)=R1
1140 LET C(50,1)=R2
1150 LET C(51,0)=R1
1160 LET C(51,1)=R2
1170 LET C(52,0)=R1
1180 LET C(52,1)=R2
1190 LET C(53,0)=R1
1200 LET C(53,1)=R2
1210 LET C(54,0)=R1
1220 LET C(54,1)=R2
1230 LET C(55,0)=R1
1240 LET C(55,1)=R2
1250 LET C(56,0)=R1
1260 LET C(56,1)=R2
1270 LET C(57,0)=R1
1280 LET C(57,1)=R2
1290 LET C(58,0)=R1
1300 LET C(58,1)=R2
1310 LET C(59,0)=R1
1320 LET C(59,1)=R2
1330 LET C(60,0)=R1
1340 LET C(60,1)=R2
1350 LET C(61,0)=R1
1360 LET C(61,1)=R2
1370 LET C(62,0)=R1
1380 LET C(62,1)=R2
1390 LET C(63,0)=R1
1400 LET C(63,1)=R2
1410 LET C(64,0)=R1
1420 LET C(64,1)=R2
1430 LET C(65,0)=R1
1440 LET C(65,1)=R2
1450 LET C(66,0)=R1
1460 LET C(66,1)=R2
1470 LET C(67,0)=R1
1480 LET C(67,1)=R2
1490 LET C(68,0)=R1
1500 LET C(68,1)=R2
1510 LET C(69,0)=R1
1520 LET C(69,1)=R2
1530 LET C(70,0)=R1
1540 LET C(70,1)=R2
1550 LET C(71,0)=R1
1560 LET C(71,1)=R2
1570 LET C(72,0)=R1
1580 LET C(72,1)=R2
1590 LET C(73,0)=R1
1600 LET C(73,1)=R2
1610 LET C(74,0)=R1
1620 LET C(74,1)=R2
1630 LET C(75,0)=R1
1640 LET C(75,1)=R2
1650 LET C(76,0)=R1
1660 LET C(76,1)=R2
1670 LET C(77,0)=R1
1680 LET C(77,1)=R2
1690 LET C(78,0)=R1
1700 LET C(78,1)=R2
1710 LET C(79,0)=R1
1720 LET C(79,1)=R2
1730 LET C(80,0)=R1
1740 LET C(80,1)=R2
1750 LET C(81,0)=R1
1760 LET C(81,1)=R2
1770 LET C(82,0)=R1
1780 LET C(82,1)=R2
1790 LET C(83,0)=R1
1800 LET C(83,1)=R2
1810 LET C(84,0)=R1
1820 LET C(84,1)=R2
1830 LET C(85,0)=R1
1840 LET C(85,1)=R2
1850 LET C(86,0)=R1
1860 LET C(86,1)=R2
1870 LET C(87,0)=R1
1880 LET C(87,1)=R2
1890 LET C(88,0)=R1
1900 LET C(88,1)=R2
1910 LET C(89,0)=R1
1920 LET C(89,1)=R2
1930 LET C(90,0)=R1
1940 LET C(90,1)=R2
1950 LET C(91,0)=R1
1960 LET C(91,1)=R2
1970 LET C(92,0)=R1
1980 LET C(92,1)=R2
1990 LET C(93,0)=R1
2000 LET C(93,1)=R2
2010 LET C(94,0)=R1
2020 LET C(94,1)=R2
2030 LET C(95,0)=R1
2040 LET C(95,1)=R2
2050 LET C(96,0)=R1
2060 LET C(96,1)=R2
2070 LET C(97,0)=R1
2080 LET C(97,1)=R2
2090 LET C(98,0)=R1
2100 LET C(98,1)=R2
2110 LET C(99,0)=R1
2120 LET C(99,1)=R2
2130 LET C(100,0)=R1
2140 LET C(100,1)=R2
2150 LET C(101,0)=R1
2160 LET C(101,1)=R2
2170 LET C(102,0)=R1
2180 LET C(102,1)=R2
2190 LET C(103,0)=R1
2200 LET C(103,1)=R2
2210 LET C(104,0)=R1
2220 LET C(104,1)=R2
2230 LET C(105,0)=R1
2240 LET C(105,1)=R2
2250 LET C(106,0)=R1
2260 LET C(106,1)=R2
2270 LET C(107,0)=R1
2280 LET C(107,1)=R2
2290 LET C(108,0)=R1
2300 LET C(108,1)=R2
2310 LET C(109,0)=R1
2320 LET C(109,1)=R2
2330 LET C(110,0)=R1
2340 LET C(110,1)=R2
2350 LET C(111,0)=R1
2360 LET C(111,1)=R2
2370 LET C(112,0)=R1
2380 LET C(112,1)=R2
2390 LET C(113,0)=R1
2400 LET C(113,1)=R2
2410 LET C(114,0)=R1
2420 LET C(114,1)=R2
2430 LET C(115,0)=R1
2440 LET C(115,1)=R2
2450 LET C(116,0)=R1
2460 LET C(116,1)=R2
2470 LET C(117,0)=R1
2480 LET C(117,1)=R2
2490 LET C(118,0)=R1
2500 LET C(118,1)=R2
2510 LET C(119,0)=R1
2520 LET C(119,1)=R2
2530 LET C(120,0)=R1
2540 LET C(120,1)=R2
2550 LET C(121,0)=R1
2560 LET C(121,1)=R2
2570 LET C(122,0)=R1
2580 LET C(122,1)=R2
2590 LET C(123,0)=R1
2600 LET C(123,1)=R2
2610 LET C(124,0)=R1
2620 LET C(124,1)=R2
2630 LET C(125,0)=R1
2640 LET C(125,1)=R2
2650 LET C(126,0)=R1
2660 LET C(126,1)=R2
2670 LET C(127,0)=R1
2680 LET C(127,1)=R2
2690 LET C(128,0)=R1
2700 LET C(128,1)=R2
2710 LET C(129,0)=R1
2720 LET C(129,1)=R2
2730 LET C(130,0)=R1
2740 LET C(130,1)=R2
2750 LET C(131,0)=R1
2760 LET C(131,1)=R2
2770 LET C(132,0)=R1
2780 LET C(132,1)=R2
2790 LET C(133,0)=R1
2800 LET C(133,1)=R2
2810 LET C(134,0)=R1
2820 LET C(134,1)=R2
2830 LET C(135,0)=R1
2840 LET C(135,1)=R2
2850 LET C(136,0)=R1
2860 LET C(136,1)=R2
2870 LET C(137,0)=R1
2880 LET C(137,1)=R2
2890 LET C(138,0)=R1
2900 LET C(138,1)=R2
2910 LET C(139,0)=R1
2920 LET C(139,1)=R2
2930 LET C(140,0)=R1
2940 LET C(140,1)=R2
2950 LET C(141,0)=R1
2960 LET C(141,1)=R2
2970 LET C(142,0)=R1
2980 LET C(142,1)=R2
2990 LET C(143,0)=R1
3000 LET C(143,1)=R2
3010 LET C(144,0)=R1
3020 LET C(144,1)=R2
3030 LET C(145,0)=R1
3040 LET C(145,1)=R2
3050 LET C(146,0)=R1
3060 LET C(146,1)=R2
3070 LET C(147,0)=R1
3080 LET C(147,1)=R2
3090 LET C(148,0)=R1
3100 LET C(148,1)=R2
3110 LET C(149,0)=R1
3120 LET C(149,1)=R2
3130 LET C(150,0)=R1
3140 LET C(150,1)=R2
3150 LET C(151,0)=R1
3160 LET C(151,1)=R2
3170 LET C(152,0)=R1
3180 LET C(152,1)=R2
3190 LET C(153,0)=R1
3200 LET C(153,1)=R2
3210 LET C(154,0)=R1
3220 LET C(154,1)=R2
3230 LET C(155,0)=R1
3240 LET C(155,1)=R2
3250 LET C(156,0)=R1
3260 LET C(156,1)=R2
3270 LET C(157,0)=R1
3280 LET C(157,1)=R2
3290 LET C(158,0)=R1
3300 LET C(158,1)=R2
3310 LET C(159,0)=R1
3320 LET C(159,1)=R2
3330 LET C(160,0)=R1
3340 LET C(160,1)=R2
3350 LET C(161,0)=R1
3360 LET C(161,1)=R2
3370 LET C(162,0)=R1
3380 LET C(162,1)=R2
3390 LET C(163,0)=R1
3400 LET C(163,1)=R2
3410 LET C(164,0)=R1
3420 LET C(164,1)=R2
3430 LET C(165,0)=R1
3440 LET C(165,1)=R2
3450 LET C(166,0)=R1
3460 LET C(166,1)=R2
3470 LET C(167,0)=R1
3480 LET C(167,1)=R2
3490 LET C(168,0)=R1
3500 LET C(168,1)=R2
3510 LET C(169,0)=R1
3520 LET C(169,1)=R2
3530 LET C(170,0)=R1
3540 LET C(170,1)=R2
3550 LET C(171,0)=R1
3560 LET C(171,1)=R2
3570 LET C(172,0)=R1
3580 LET C(172,1)=R2
3590 LET C(173,0)=R1
3600 LET C(173,1)=R2
3610 LET C(174,0)=R1
3620 LET C(174,1)=R2
3630 LET C(175,0)=R1
3640 LET C(175,1)=R2
3650 LET C(176,0)=R1
3660 LET C(176,1)=R2
3670 LET C(177,0)=R1
3680 LET C(177,1)=R2
3690 LET C(178,0)=R1
3700 LET C(178,1)=R2
3710 LET C(179,0)=R1
3720 LET C(179,1)=R2
3730 LET C(180,0)=R1
3740 LET C(180,1)=R2
3750 LET C(181,0)=R1
3760 LET C(181,1)=R2
3770 LET C(182,0)=R1
3780 LET C(182,1)=R2
3790 LET C(183,0)=R1
3800 LET C(183,1)=R2
3810 LET C(184,0)=R1
3820 LET C(184,1)=R2
3830 LET C(185,0)=R1
3840 LET C(185,1)=R2
3850 LET C(186,0)=R1
3860 LET C(186,1)=R2
3870 LET C(187,0)=R1
3880 LET C(187,1)=R2
3890 LET C(188,0)=R1
3900 LET C(188,1)=R2
3910 LET C(189,0)=R1
3920 LET C(189,1)=R2
3930 LET C(190,0)=R1
3940 LET C(190,1)=R2
3950 LET C(191,0)=R1
3960 LET C(191,1)=R2
3970 LET C(192,0)=R1
3980 LET C(192,1)=R2
3990 LET C(193,0)=R1
4000 LET C(193,1)=R2
4010 LET C(194,0)=R1
4020 LET C(194,1)=R2
4030 LET C(195,0)=R1
4040 LET C(195,1)=R2
4050 LET C(196,0)=R1
4060 LET C(196,1)=R2
4070 LET C(197,0)=R1
4080 LET C(197,1)=R2
4090 LET C(198,0)=R1
4100 LET C(198,1)=R2
4110 LET C(199,0)=R1
4120 LET C(199,1)=R2
4130 LET C(200,0)=R1
4140 LET C(200,1)=R2
4150 LET C(201,0)=R1
4160 LET C(201,1)=R2
4170 LET C(202,0)=R1
4180 LET C(202,1)=R2
4190 LET C(203,0)=R1
4200 LET C(203,1)=R2
4210 LET C(204,0)=R1
4220 LET C(204,1)=R2
4230 LET C(205,0)=R1
4240 LET C(205,1)=R2
4250 LET C(206,0)=R1
4260 LET C(206,1)=R2
4270 LET C(207,0)=R1
4280 LET C(207,1)=R2
4290 LET C(208,0)=R1
4300 LET C(208,1)=R2
4310 LET C(209,0)=R1
4320 LET C(209,1)=R2
4330 LET C(210,0)=R1
4340 LET C(210,1)=R2
4350 LET C(211,0)=R1
4360 LET C(211,1)=R2
4370 LET C(212,0)=R1
4380 LET C(212,1)=R2
4390 LET C(213,0)=R1
4400 LET C(213,1)=R2
4410 LET C(214,0)=R1
4420 LET C(214,1)=R2
4430 LET C(215,0)=R1
4440 LET C(215,1)=R2
4450 LET C(216,0)=R1
4460 LET C(216,1)=R2
4470 LET C(217,0)=R1
4480 LET C(217,1)=R2
4490 LET C(218,0)=R1
4500 LET C(218,1)=R2
4510 LET C(219,0)=R1
4520 LET C(219,1)=R2
4530 LET C(220,0)=R1
4540 LET C(220,1)=R2
4550 LET C(221,0)=R1
4560 LET C(221,1)=R2
4570 LET C(222,0)=R1
4580 LET C(222,1)=R2
4590 LET C(223,0)=R1
4600 LET C(223,1)=R2
4610 LET C(224,0)=R1
4620 LET C(224,1)=R2
4630 LET C(225,0)=R1
4640 LET C(225,1)=R2
4650 LET C(226,0)=R1
4660 LET C(226,1)=R2
4670 LET C(227,0)=R1
4680 LET C(227,1)=R2
4690 LET C(228,0)=R1
4700 LET C(228,1)=R2
4710 LET C(229,0)=R1
4720 LET C(229,1)=R2
4730 LET C(230,0)=R1
4740 LET C(230,1)=R2
4750 LET C(231,0)=R1
4760 LET C(231,1)=R2
4770 LET C(232,0)=R1
4780 LET C(232,1)=R2
4790 LET C(233,0)=R1
4800 LET C(233,1)=R2
4810 LET C(234,0)=R1
4820 LET C(234,1)=R2
4830 LET C(235,0)=R1
4840 LET C(235,1)=R2
4850 LET C(236,0)=R1
4860 LET C(236,1)=R2
4870 LET C(237,0)=R1
4880 LET C(237,1)=R2
4890 LET C(238,0)=R1
4900 LET C(238,1)=R2
4910 LET C(239,0)=R1
4920 LET C(239,1)=R2
4930 LET C(240,0)=R1
4940 LET C(240,1)=R2
4950 LET C(241,0)=R1
4960 LET C(241,1)=R2
4970 LET C(242,0)=R1
4980 LET C(242,1)=R2
4990 LET C(243,0)=R1
5000 LET C(243,1)=R2
5010 LET C(244,0)=R1
5020 LET C(244,1)=R2
5030 LET C(245,0)=R1
5040 LET C(245,1)=R2
5050 LET C(246,0)=R1
5060 LET C(246,1)=R2
5070 LET C(247,0)=R1
5080 LET C(247,1)=R2
5090 LET C(248,0)=R1
5100 LET C(248,1)=R2
5110 LET C(249,0)=R1
5120 LET C(249,1)=R2
5130 LET C(250,0)=R1
5140 LET C(250,1)=R2
5150 LET C(251,0)=R1
5160 LET C(251,1)=R2
5170 LET C(252,0)=R1
5180 LET C(252,1)=R2
5190 LET C(253,0)=R1
5200 LET C(253,1)=R2
5210 LET C(254,0)=R1
5220 LET C(254,1)=R2
5230 LET C(255,0)=R1
5240 LET C(255,1)=R2
5250 LET C(256,0)=R1
5260 LET C(256,1)=R2
5270 LET C(257,0)=R1
5280 LET C(257,1)=R2
5290 LET C(258,0)=R1
5300 LET C(258,1)=R2
5310 LET C(259,0)=R1
5320 LET C(259,1)=R2
5330 LET C(260,0)=R1
5340 LET C(260,1)=R2
5350 LET C(261,0)=R1
5360 LET C(261,1)=R2
5370 LET C(262,0)=R1
5380 LET C(262,1)=R2
5390 LET C(263,0)=R1
5400 LET C(263,1)=R2
5410 LET C(264,0)=R1
5420 LET C(264,1)=R2
5430 LET C(265,0)=R1
5440 LET C(265,1)=R2
5450 LET C(266,0)=R1
5460 LET C(266,1)=R2
5470 LET C(267,0)=R1
5480 LET C(267,1)=R2
5490 LET C(268,0)=R1
5500 LET C(268,1)=R2
5510 LET C(269,0)=R1
5520 LET C(269,1)=R2
5530 LET C(270,0)=R1
5540 LET C(270,1)=R2
5550 LET C(271,0)=R1
5560 LET C(271,1)=R2
5570 LET C(272,0)=R1
5580 LET C(272,1)=R2
5590 LET C(273,0)=R1
5600 LET C(273,1)=R2
5610 LET C(274,0)=R1
5620 LET C(274,1)=R2
5630 LET C(275,0)=R1
5640 LET C(275,1)=R2
5650 LET C(276,0)=R1
5660 LET C(276,1)=R2
5670 LET C(277,0)=R1
5680 LET C(277,1)=R2
5690 LET C(278,0)=R1
5700 LET C(278,1)=R2
5710 LET C(279,0)=R1
5720 LET C(279,1)=R2
5730 LET C(280,0)=R1
5740 LET C(280,1)=R2
5750 LET C(281,0)=R1
5760 LET C(281,1)=R2
5770 LET C(282,0)=R1
5780 LET C(282,1)=R2
5790 LET C(283,0)=R1
5800 LET C(283,1)=R2
5810 LET C(284,0)=R1
5820 LET C(284,1)=R2
5830 LET C(285,0)=R1
5840 LET C(285,1)=R2
5850 LET C(286,0)=R1
5860 LET C(286,1)=R2
5870 LET C(287,0)=R1
5880 LET C(287,1)=R2
5890 LET C(288,0)=R1
5900 LET C(288,1)=R2
5910 LET C(289,0)=R1
5920 LET C(289,1)=R2
5930 LET C(290,0)=R1
5940 LET C(290,1)=R2
5950 LET C(291,0)=R1
5960 LET C(291,1)=R2
5970 LET C(292,0)=R1
5980 LET C(292,1)=R2
5990 LET C(293,0)=R1
6000 LET C(293,1)=R2
6010 LET C(294,0)=R1
6020 LET C(294,1)=R2
6030 LET C(295,0)=R1
6040 LET C(295,1)=R2
6050 LET C(296,0)=R1
6060 LET C(296,1)=R2
6070 LET C(297,0)=R1
6080 LET C(297,1)=R2
6090 LET C(298,0)=R1
6100 LET C(298,1)=R2
6110 LET C(299,0)=R1
6120 LET C(299,1)=R2
6130 LET C(300,0)=R1
6140 LET C(300,1)=R2
6150 LET C(301,0)=R1
6160 LET C(301,1)=R2
6170 LET C(302,0)=R1
6180 LET C(302,1)=R2
6190 LET C(303,0)=R1
6200 LET C(303,1)=R2
6210 LET C(304,0)=R1
6220 LET C(304,1)=R2
6230 LET C(305,0)=R1
6240 LET C(305,1)=R2
6250 LET C(306,0)=R1
6260 LET C(306,1)=R2
6270 LET C(307,0)=R1
6280 LET C(307,1)=R2
6290 LET C(308,0)=R1
6300 LET C(308,1)=R2
6310 LET C(309,0)=R1
6320 LET C(309,1)=R2
6330 LET C(310,0)=R1
6340 LET C(310,1)=R2
6350 LET C(311,0)=R1
6360 LET C(311,1)=R2
6370 LET C(312,0)=R1
6380 LET C(312,1)=R2
6390 LET C(313,0)=R1
6400 LET C(313,1)=R2
6410 LET C(314,0)=R1
6420 LET C(314,1)=R2
6430 LET C(315,0)=R1
6440 LET C(315,1)=R2
6450 LET C(316,0)=R1
6460 LET C(316,1)=R2
6470 LET C(317,0)=R1
6480 LET C(317,1)=R2
6490 LET C(318,0)=R1
6500 LET C(318,1)=R2
6510 LET C(319,0)=R1
6520 LET C(319,1)=R2
6530 LET C(320,0)=R1
6540 LET C(320,1)=R2
6550 LET C(321,0)=R1
6560 LET C(321,1)=R2
6570 LET C(322,0)=R1
6580 LET C(322,1)=R2
6590 LET C(323,0)=R1
6600 LET C(323,1)=R2
6610 LET C(324,0)=R1
6620 LET C(324,1)=R2
6630 LET C(325,0)=R1
6640 LET C(325,1)=R2
6650 LET C(326,0)=R1
6660 LET C(326,1)=R2
6670 LET C(327,0)=R1
6680 LET C(327,1)=R2
6690 LET C(328,0)=R1
6700 LET C(328,1)=R2
6710 LET C(329,0)=R1
6720 LET C(329,1)=R2
6730 LET C(330,0)=R1
6740 LET C(330,1)=R2
6750 LET C(331,0)=R1
6760 LET C(331,1)=R2
6770 LET C(332,0)=R1
6780 LET C(332,1)=R2
6790 LET C(333,0)=R1
6800 LET C(333,1)=R2
6810 LET C(334,0)=R1
6820 LET C(334,1)=R2
6830 LET C(335,0)=R1
6840 LET C(335,1)=R2
6850 LET C(336,0)=R1
6860 LET C(336,1)=R2
6870 LET C(337,0)=R1
6880 LET C(337,1)=R2
6890 LET C(338,0)=R1
6900 LET C(338,1)=R2
6910 LET C(339,0)=R1
6920 LET C(339,1)=R2
6930 LET C(340,0)=R1
6940 LET C(340,1)=R2
6950 LET C(341,0)=R1
6960 LET C(341,1)=R2
6970 LET C(342,0)=R1
6980 LET C(342,1)=R2
6990 LET C(343,0)=R1
7000 LET C(343,1)=R2
7010 LET C(344,0)=R1
7020 LET C(344,1)=R2
7030 LET C(345,0)=R1
7040 LET C(345,1)=R2
7050 LET C(346,0)=R1
7060 LET C(346,1)=R2
7070 LET C(347,0)=R1
7080 LET C(347,1)=R2
7090 LET C(348,0)=R1
7100 LET C(348,1)=R2
7110 LET C(349,0)=R1
7120 LET C(349,1)=R2
7130 LET C(350,0)=R1
7140 LET C(350,1)=R2
7150 LET C(351,0)=R1
7160 LET C(351,1)=R2
7170 LET C(352,0)=R1
7180 LET C(352,1)=R2
7190 LET C(353,0)=R1
7200 LET C(353,1
```

```

0000 IF F=4 THEN LET T(4)=T(1)
0001 IF F=5 THEN
0002 PRINT AT L,5: "YOUR MOVE PRO
N 1
0003 INPUT G
0004 LET C=DOOR-DB-37
0005 IF C=0 THEN GOTO 1000
0006 IF C=1 THEN GOTO 1000
0007 IF C=2 THEN GOTO 1000
0008 PRINT G: " TO
0009 INPUT H
0010 IF H=0 THEN THEN PRINT AT L,
1
0011 IF H=0 THEN GOTO 370
0012 LET C=DOOR-DB-37
0013 INPUT H
0014 IF C=0 THEN THEN GOTO 1000
0015 IF H=0 THEN THEN GOTO 1000
0016 IF C=1 THEN GOTO 1000
0017 IF C=2 THEN GOTO 1000
0018 IF C=3 THEN GOTO 1000
0019 IF C=4 THEN GOTO 1000
0020 IF C=5 THEN GOTO 1000
0021 IF C=6 THEN GOTO 1000
0022 IF C=7 THEN GOTO 1000
0023 IF C=8 THEN GOTO 1000
0024 IF C=9 THEN GOTO 1000
0025 IF C=10 THEN GOTO 1000
0026 IF C=11 THEN GOTO 1000
0027 IF C=12 THEN GOTO 1000
0028 IF C=13 THEN GOTO 1000
0029 IF C=14 THEN GOTO 1000
0030 IF C=15 THEN GOTO 1000
0031 IF C=16 THEN GOTO 1000
0032 IF C=17 THEN GOTO 1000
0033 IF C=18 THEN GOTO 1000
0034 IF C=19 THEN GOTO 1000
0035 IF C=20 THEN GOTO 1000
0036 IF C=21 THEN GOTO 1000
0037 IF C=22 THEN GOTO 1000
0038 IF C=23 THEN GOTO 1000
0039 IF C=24 THEN GOTO 1000
0040 IF C=25 THEN GOTO 1000
0041 IF C=26 THEN GOTO 1000
0042 IF C=27 THEN GOTO 1000
0043 IF C=28 THEN GOTO 1000
0044 IF C=29 THEN GOTO 1000
0045 IF C=30 THEN GOTO 1000
0046 IF C=31 THEN GOTO 1000
0047 IF C=32 THEN GOTO 1000
0048 IF C=33 THEN GOTO 1000
0049 IF C=34 THEN GOTO 1000
0050 IF C=35 THEN GOTO 1000
0051 IF C=36 THEN GOTO 1000
0052 IF C=37 THEN GOTO 1000
0053 IF C=38 THEN GOTO 1000
0054 IF C=39 THEN GOTO 1000
0055 IF C=40 THEN GOTO 1000
0056 IF C=41 THEN GOTO 1000
0057 IF C=42 THEN GOTO 1000
0058 IF C=43 THEN GOTO 1000
0059 IF C=44 THEN GOTO 1000
0060 IF C=45 THEN GOTO 1000
0061 IF C=46 THEN GOTO 1000
0062 IF C=47 THEN GOTO 1000
0063 IF C=48 THEN GOTO 1000
0064 IF C=49 THEN GOTO 1000
0065 IF C=50 THEN GOTO 1000
0066 IF C=51 THEN GOTO 1000
0067 IF C=52 THEN GOTO 1000
0068 IF C=53 THEN GOTO 1000
0069 IF C=54 THEN GOTO 1000
0070 IF C=55 THEN GOTO 1000
0071 IF C=56 THEN GOTO 1000
0072 IF C=57 THEN GOTO 1000
0073 IF C=58 THEN GOTO 1000
0074 IF C=59 THEN GOTO 1000
0075 IF C=60 THEN GOTO 1000
0076 IF C=61 THEN GOTO 1000
0077 IF C=62 THEN GOTO 1000
0078 IF C=63 THEN GOTO 1000
0079 IF C=64 THEN GOTO 1000
0080 IF C=65 THEN GOTO 1000
0081 IF C=66 THEN GOTO 1000
0082 IF C=67 THEN GOTO 1000
0083 IF C=68 THEN GOTO 1000
0084 IF C=69 THEN GOTO 1000
0085 IF C=70 THEN GOTO 1000
0086 IF C=71 THEN GOTO 1000
0087 IF C=72 THEN GOTO 1000
0088 IF C=73 THEN GOTO 1000
0089 IF C=74 THEN GOTO 1000
0090 IF C=75 THEN GOTO 1000
0091 IF C=76 THEN GOTO 1000
0092 IF C=77 THEN GOTO 1000
0093 IF C=78 THEN GOTO 1000
0094 IF C=79 THEN GOTO 1000
0095 IF C=80 THEN GOTO 1000
0096 IF C=81 THEN GOTO 1000
0097 IF C=82 THEN GOTO 1000
0098 IF C=83 THEN GOTO 1000
0099 IF C=84 THEN GOTO 1000
0100 IF C=85 THEN GOTO 1000
0101 IF C=86 THEN GOTO 1000
0102 IF C=87 THEN GOTO 1000
0103 IF C=88 THEN GOTO 1000
0104 IF C=89 THEN GOTO 1000
0105 IF C=90 THEN GOTO 1000
0106 IF C=91 THEN GOTO 1000
0107 IF C=92 THEN GOTO 1000
0108 IF C=93 THEN GOTO 1000
0109 IF C=94 THEN GOTO 1000
0110 IF C=95 THEN GOTO 1000
0111 IF C=96 THEN GOTO 1000
0112 IF C=97 THEN GOTO 1000
0113 IF C=98 THEN GOTO 1000
0114 IF C=99 THEN GOTO 1000
0115 IF C=100 THEN GOTO 1000
0116 IF C=101 THEN GOTO 1000
0117 IF C=102 THEN GOTO 1000
0118 IF C=103 THEN GOTO 1000
0119 IF C=104 THEN GOTO 1000
0120 IF C=105 THEN GOTO 1000
0121 IF C=106 THEN GOTO 1000
0122 IF C=107 THEN GOTO 1000
0123 IF C=108 THEN GOTO 1000
0124 IF C=109 THEN GOTO 1000
0125 IF C=110 THEN GOTO 1000
0126 IF C=111 THEN GOTO 1000
0127 IF C=112 THEN GOTO 1000
0128 IF C=113 THEN GOTO 1000
0129 IF C=114 THEN GOTO 1000
0130 IF C=115 THEN GOTO 1000
0131 IF C=116 THEN GOTO 1000
0132 IF C=117 THEN GOTO 1000
0133 IF C=118 THEN GOTO 1000
0134 IF C=119 THEN GOTO 1000
0135 IF C=120 THEN GOTO 1000
0136 IF C=121 THEN GOTO 1000
0137 IF C=122 THEN GOTO 1000
0138 IF C=123 THEN GOTO 1000
0139 IF C=124 THEN GOTO 1000
0140 IF C=125 THEN GOTO 1000
0141 IF C=126 THEN GOTO 1000
0142 IF C=127 THEN GOTO 1000
0143 IF C=128 THEN GOTO 1000
0144 IF C=129 THEN GOTO 1000
0145 IF C=130 THEN GOTO 1000
0146 IF C=131 THEN GOTO 1000
0147 IF C=132 THEN GOTO 1000
0148 IF C=133 THEN GOTO 1000
0149 IF C=134 THEN GOTO 1000
0150 IF C=135 THEN GOTO 1000
0151 IF C=136 THEN GOTO 1000
0152 IF C=137 THEN GOTO 1000
0153 IF C=138 THEN GOTO 1000
0154 IF C=139 THEN GOTO 1000
0155 IF C=140 THEN GOTO 1000
0156 IF C=141 THEN GOTO 1000
0157 IF C=142 THEN GOTO 1000
0158 IF C=143 THEN GOTO 1000
0159 IF C=144 THEN GOTO 1000
0160 IF C=145 THEN GOTO 1000
0161 IF C=146 THEN GOTO 1000
0162 IF C=147 THEN GOTO 1000
0163 IF C=148 THEN GOTO 1000
0164 IF C=149 THEN GOTO 1000
0165 IF C=150 THEN GOTO 1000
0166 IF C=151 THEN GOTO 1000
0167 IF C=152 THEN GOTO 1000
0168 IF C=153 THEN GOTO 1000
0169 IF C=154 THEN GOTO 1000
0170 IF C=155 THEN GOTO 1000
0171 IF C=156 THEN GOTO 1000
0172 IF C=157 THEN GOTO 1000
0173 IF C=158 THEN GOTO 1000
0174 IF C=159 THEN GOTO 1000
0175 IF C=160 THEN GOTO 1000
0176 IF C=161 THEN GOTO 1000
0177 IF C=162 THEN GOTO 1000
0178 IF C=163 THEN GOTO 1000
0179 IF C=164 THEN GOTO 1000
0180 IF C=165 THEN GOTO 1000
0181 IF C=166 THEN GOTO 1000
0182 IF C=167 THEN GOTO 1000
0183 IF C=168 THEN GOTO 1000
0184 IF C=169 THEN GOTO 1000
0185 IF C=170 THEN GOTO 1000
0186 IF C=171 THEN GOTO 1000
0187 IF C=172 THEN GOTO 1000
0188 IF C=173 THEN GOTO 1000
0189 IF C=174 THEN GOTO 1000
0190 IF C=175 THEN GOTO 1000
0191 IF C=176 THEN GOTO 1000
0192 IF C=177 THEN GOTO 1000
0193 IF C=178 THEN GOTO 1000
0194 IF C=179 THEN GOTO 1000
0195 IF C=180 THEN GOTO 1000
0196 IF C=181 THEN GOTO 1000
0197 IF C=182 THEN GOTO 1000
0198 IF C=183 THEN GOTO 1000
0199 IF C=184 THEN GOTO 1000
0200 IF C=185 THEN GOTO 1000
0201 IF C=186 THEN GOTO 1000
0202 IF C=187 THEN GOTO 1000
0203 IF C=188 THEN GOTO 1000
0204 IF C=189 THEN GOTO 1000
0205 IF C=190 THEN GOTO 1000
0206 IF C=191 THEN GOTO 1000
0207 IF C=192 THEN GOTO 1000
0208 IF C=193 THEN GOTO 1000
0209 IF C=194 THEN GOTO 1000
0210 IF C=195 THEN GOTO 1000
0211 IF C=196 THEN GOTO 1000
0212 IF C=197 THEN GOTO 1000
0213 IF C=198 THEN GOTO 1000
0214 IF C=199 THEN GOTO 1000
0215 IF C=200 THEN GOTO 1000
0216 IF C=201 THEN GOTO 1000
0217 IF C=202 THEN GOTO 1000
0218 IF C=203 THEN GOTO 1000
0219 IF C=204 THEN GOTO 1000
0220 IF C=205 THEN GOTO 1000
0221 IF C=206 THEN GOTO 1000
0222 IF C=207 THEN GOTO 1000
0223 IF C=208 THEN GOTO 1000
0224 IF C=209 THEN GOTO 1000
0
```

```

0110 IF B=0 THEN GOTO 0120
0120 IF C=0 THEN GOTO 0130
0130 NEXT N
0140 FOR M=1 TO 30
0150 IF B=0 THEN NEXT M
0160 IF B=1 THEN NEXT M
0170 IF B=2 THEN STOP
0180 IF B=3 THEN GOTO 0190
0190 IF C=0 THEN GOTO 0140
0200 NEXT M
0210 IF C=0 THEN GOTO 0220
0220 IF C=1 THEN GOTO 0230
0230 IF C=2 THEN GOTO 0240
0240 IF C=3 THEN GOTO 0250
0250 IF B=0 THEN GOTO 0260
0260 IF B=1 THEN GOTO 0270
0270 IF B=2 THEN GOTO 0280
0280 IF B=3 THEN GOTO 0290
0290 IF C=0 THEN GOTO 0300
0300 IF C=1 THEN GOTO 0310
0310 IF C=2 THEN GOTO 0320
0320 IF C=3 THEN GOTO 0330
0330 IF B=0 THEN GOTO 0340
0340 IF B=1 THEN GOTO 0350
0350 IF B=2 THEN GOTO 0360
0360 IF B=3 THEN GOTO 0370
0370 IF C=0 THEN GOTO 0380
0380 IF C=1 THEN GOTO 0390
0390 IF C=2 THEN GOTO 0400
0400 IF C=3 THEN GOTO 0410
0410 IF B=0 THEN GOTO 0420
0420 IF B=1 THEN GOTO 0430
0430 IF B=2 THEN GOTO 0440
0440 IF B=3 THEN GOTO 0450
0450 IF C=0 THEN GOTO 0460
0460 IF C=1 THEN GOTO 0470
0470 IF C=2 THEN GOTO 0480
0480 IF C=3 THEN GOTO 0490
0490 IF B=0 THEN GOTO 0500
0500 IF B=1 THEN GOTO 0510
0510 IF B=2 THEN GOTO 0520
0520 IF B=3 THEN GOTO 0530
0530 IF C=0 THEN GOTO 0540
0540 IF C=1 THEN GOTO 0550
0550 IF C=2 THEN GOTO 0560
0560 IF C=3 THEN GOTO 0570
0570 IF B=0 THEN GOTO 0580
0580 IF B=1 THEN GOTO 0590
0590 IF B=2 THEN GOTO 0600
0600 IF B=3 THEN GOTO 0610
0610 IF C=0 THEN GOTO 0620
0620 IF C=1 THEN GOTO 0630
0630 IF C=2 THEN GOTO 0640
0640 IF C=3 THEN GOTO 0650
0650 IF B=0 THEN GOTO 0660
0660 IF B=1 THEN GOTO 0670
0670 IF B=2 THEN GOTO 0680
0680 IF B=3 THEN GOTO 0690
0690 IF C=0 THEN GOTO 0700
0700 IF C=1 THEN GOTO 0710
0710 IF C=2 THEN GOTO 0720
0720 IF C=3 THEN GOTO 0730
0730 IF B=0 THEN GOTO 0740
0740 IF B=1 THEN GOTO 0750
0750 IF B=2 THEN GOTO 0760
0760 IF B=3 THEN GOTO 0770
0770 IF C=0 THEN GOTO 0780
0780 IF C=1 THEN GOTO 0790
0790 IF C=2 THEN GOTO 0800
0800 IF C=3 THEN GOTO 0810
0810 IF B=0 THEN GOTO 0820
0820 IF B=1 THEN GOTO 0830
0830 IF B=2 THEN GOTO 0840
0840 IF B=3 THEN GOTO 0850
0850 IF C=0 THEN GOTO 0860
0860 IF C=1 THEN GOTO 0870
0870 IF C=2 THEN GOTO 0880
0880 IF C=3 THEN GOTO 0890
0890 IF B=0 THEN GOTO 0900
0900 IF B=1 THEN GOTO 0910
0910 IF B=2 THEN GOTO 0920
0920 IF B=3 THEN GOTO 0930
0930 IF C=0 THEN GOTO 0940
0940 IF C=1 THEN GOTO 0950
0950 IF C=2 THEN GOTO 0960
0960 IF C=3 THEN GOTO 0970
0970 IF B=0 THEN GOTO 0980
0980 IF B=1 THEN GOTO 0990
0990 IF B=2 THEN GOTO 1000
1000 IF B=3 THEN GOTO 1010
1010 IF C=0 THEN GOTO 1020
1020 IF C=1 THEN GOTO 1030
1030 IF C=2 THEN GOTO 1040
1040 IF C=3 THEN GOTO 1050
1050 IF B=0 THEN GOTO 1060
1060 IF B=1 THEN GOTO 1070
1070 IF B=2 THEN GOTO 1080
1080 IF B=3 THEN GOTO 1090
1090 IF C=0 THEN GOTO 1100
1100 IF C=1 THEN GOTO 1110
1110 IF C=2 THEN GOTO 1120
1120 IF C=3 THEN GOTO 1130
1130 IF B=0 THEN GOTO 1140
1140 IF B=1 THEN GOTO 1150
1150 IF B=2 THEN GOTO 1160
1160 IF B=3 THEN GOTO 1170
1170 IF C=0 THEN GOTO 1180
1180 IF C=1 THEN GOTO 1190
1190 IF C=2 THEN GOTO 1200
1200 IF C=3 THEN GOTO 1210
1210 IF B=0 THEN GOTO 1220
1220 IF B=1 THEN GOTO 1230
1230 IF B=2 THEN GOTO 1240
1240 IF B=3 THEN GOTO 1250
1250 IF C=0 THEN GOTO 1260
1260 IF C=1 THEN GOTO 1270
1270 IF C=2 THEN GOTO 1280
1280 IF C=3 THEN GOTO 1290
1290 IF B=0 THEN GOTO 1300
1300 IF B=1 THEN GOTO 1310
1310 IF B=2 THEN GOTO 1320
1320 IF B=3 THEN GOTO 1330
1330 IF C=0 THEN GOTO 1340
1340 IF C=1 THEN GOTO 1350
1350 IF C=2 THEN GOTO 1360
1360 IF C=3 THEN GOTO 1370
1370 IF B=0 THEN GOTO 1380
1380 IF B=1 THEN GOTO 1390
1390 IF B=2 THEN GOTO 1400
1400 IF B=3 THEN GOTO 1410
1410 IF C=0 THEN GOTO 1420
1420 IF C=1 THEN GOTO 1430
1430 IF C=2 THEN GOTO 1440
1440 IF C=3 THEN GOTO 1450
1450 IF B=0 THEN GOTO 1460
1460 IF B=1 THEN GOTO 1470
1470 IF B=2 THEN GOTO 1480
1480 IF B=3 THEN GOTO 1490
1490 IF C=0 THEN GOTO 1500
1500 IF C=1 THEN GOTO 1510
1510 IF C=2 THEN GOTO 1520
1520 IF C=3 THEN GOTO 1530
1530 IF B=0 THEN GOTO 1540
1540 IF B=1 THEN GOTO 1550
1550 IF B=2 THEN GOTO 1560
1560 IF B=3 THEN GOTO 1570
1570 IF C=0 THEN GOTO 1580
1580 IF C=1 THEN GOTO 1590
1590 IF C=2 THEN GOTO 1600
1600 IF C=3 THEN GOTO 1610
1610 IF B=0 THEN GOTO 1620
1620 IF B=1 THEN GOTO 1630
1630 IF B=2 THEN GOTO 1640
1640 IF B=3 THEN GOTO 1650
1650 IF C=0 THEN GOTO 1660
1660 IF C=1 THEN GOTO 1670
1670 IF C=2 THEN GOTO 1680
1680 IF C=3 THEN GOTO 1690
1690 IF B=0 THEN GOTO 1700
1700 IF B=1 THEN GOTO 1710
1710 IF B=2 THEN GOTO 1720
1720 IF B=3 THEN GOTO 1730
1730 IF C=0 THEN GOTO 1740
1740 IF C=1 THEN GOTO 1750
1750 IF C=2 THEN GOTO 1760
1760 IF C=3 THEN GOTO 1770
1770 IF B=0 THEN GOTO 1780
1780 IF B=1 THEN GOTO 1790
1790 IF B=2 THEN GOTO 1800
1800 IF B=3 THEN GOTO 1810
1810 IF C=0 THEN GOTO 1820
1820 IF C=1 THEN GOTO 1830
1830 IF C=2 THEN GOTO 1840
1840 IF C=3 THEN GOTO 1850
1850 IF B=0 THEN GOTO 1860
1860 IF B=1 THEN GOTO 1870
1870 IF B=2 THEN GOTO 1880
1880 IF B=3 THEN GOTO 1890
1890 IF C=0 THEN GOTO 1900
1900 IF C=1 THEN GOTO 1910
1910 IF C=2 THEN GOTO 1920
1920 IF C=3 THEN GOTO 1930
1930 IF B=0 THEN GOTO 1940
1940 IF B=1 THEN GOTO 1950
1950 IF B=2 THEN GOTO 1960
1960 IF B=3 THEN GOTO 1970
1970 IF C=0 THEN GOTO 1980
1980 IF C=1 THEN GOTO 1990
1990 IF C=2 THEN GOTO 2000
2000 IF C=3 THEN GOTO 2010
2010 IF B=0 THEN GOTO 2020
2020 IF B=1 THEN GOTO 2030
2030 IF B=2 THEN GOTO 2040
2040 IF B=3 THEN GOTO 2050
2050 IF C=0 THEN GOTO 2060
2060 IF C=1 THEN GOTO 2070
2070 IF C=2 THEN GOTO 2080
2080 IF C=3 THEN GOTO 2090
2090 IF B=0 THEN GOTO 2100
2100 IF B=1 THEN GOTO 2110
2110 IF B=2 THEN GOTO 2120
2120 IF B=3 THEN GOTO 2130
2130 IF C=0 THEN GOTO 2140
2140 IF C=1 THEN GOTO 2150
2150 IF C=2 THEN GOTO 2160
2160 IF C=3 THEN GOTO 2170
2170 IF B=0 THEN GOTO 2180
2180 IF B=1 THEN GOTO 2190
2190 IF B=2 THEN GOTO 2200
2200 IF B=3 THEN GOTO 2210
2210 IF C=0 THEN GOTO 2220
2220 IF C=1 THEN GOTO 2230
2230 IF C=2 THEN GOTO 2240
2240 IF C=3 THEN GOTO 2250
2250 IF B=0 THEN GOTO 2260
2260 IF B=1 THEN GOTO 2270
2270 IF B=2 THEN GOTO 2280
2280 IF B=3 THEN GOTO 2290
2290 IF C=0 THEN GOTO 2300
2300 IF C=1 THEN GOTO 2310
2310 IF C=2 THEN GOTO 2320
2320 IF C=3 THEN GOTO 2330
2330 IF B=0 THEN GOTO 2340
2340 IF B=1 THEN GOTO 2350
2350 IF B=2 THEN GOTO 2360
2360 IF B=3 THEN GOTO 2370
2370 IF C=0 THEN GOTO 2380
2380 IF C=1 THEN GOTO 2390
2390 IF C=2 THEN GOTO 2400
2400 IF C=3 THEN GOTO 2410
2410 IF B=0 THEN GOTO 2420
2420 IF B=1 THEN GOTO 2430
2430 IF B=2 THEN GOTO 2440
2440 IF B=3 THEN GOTO 2450
2450 IF C=0 THEN GOTO 2460
2460 IF C=1 THEN GOTO 2470
2470 IF C=2 THEN GOTO 2480
2480 IF C=3 THEN GOTO 2490
2490 IF B=0 THEN GOTO 2500
2500 IF B=1 THEN GOTO 2510
2510 IF B=2 THEN GOTO 2520
2520 IF B=3 THEN GOTO 2530
2530 IF C=0 THEN GOTO 2540
25
```

```

3550 IF M131=0 THEN RETURN
3560 J=J+1
3570 IF J=6 THEN POINT AT J+6,28
3580 IF J=6 AND J13 THEN PRINT
3590 IF J=6 THEN PRINT AT J-4,2
3600
3610 NEXT J
3620 RETURN
3630 IF N=0.24 THEN RETURN
3640 LET SIN=SIN(1-1
3650 LET SIN=COS(N-0.1
3660 POINT M5,CHRS (M37); TO
3670 CHRS (M-0.37)
3680 LET C=0
3690 RETURN
3700 IF N=0.24 THEN RETURN
3710 LET SIN=SIN(1-1
3720 LET SIN=COS(N-0.1
3730 LET M131=M131+1
3740 LET M131=M131+1
3750 POINT M5,CHRS (M37); TO
3760 CHRS (M-0.37); - BAR
3770 LET C=C+1
3780 RETURN
3790 IF N=0.24 THEN RETURN
3800 LET SIN=SIN(1-1
3810 LET SIN=COS(N-0.1
3820 LET M131=M131+1
3830 LET M131=M131+1
3840 POINT M5,CHRS (M37); TO
3850 CHRS (M-0.37); - BAR
3860 LET C=C+1
3870 RETURN
3880 IF M131=0 THEN RETURN
3890 LET SIN=SIN(1-1
3900 LET SIN=COS(N-0.1
3910 LET M131=M131+1
3920 LET M131=M131+1
3930 POINT M5,CHRS (M37); TO
3940 CHRS (M-0.37)
3950 LET C=C+1
3960 RETURN
3970 IF M131=0 THEN GOTO 3980
3980 IF M131=0 THEN GOTO 3970
3990 IF C=0 AND D=0 AND M131=0
4000 LET CC=CC+1
4010 IF CC=3 THEN PRINT "I HAVE
4020 FOUND A WAY TO MOVE"
4030 NEXT C
4040 GOSUB 3100
4050 LET CC=0
4060
4070 TO PRINT AT 0.0,"PRESS ANY KEY
4080 TO END"
4090 SLOW
4100 IF INKEY="" THEN GOTO 3810
4110 PRINT
4120 CLS
4130 GOTO 310
4140 LET M131=M131-1
4150 LET M131=M131+1
4160 POINT M5,BAR TO ";CHRS (0
4170
4180 IF M131=0 THEN PRINT
4190 IF M131=1 THEN GOSUB 3030
4200 LET C=C+1
4210 IF M131=0 THEN GOTO 3100
4220 GOTO 1000
4230 LET M131=M131-1
4240 LET M131=M131+1
4250 PRINT - BAR
4260 RETURN
4270 LET M131=M131-1
4280 LET M131=M131+1
4290 POINT M5,BAR TO ";CHRS (0
4300
4310 IF M131=0 THEN PRINT
4320 IF M131=1 THEN GOSUB 3140
4330 LET C=C+1
4340 GOTO 1000
4350 LET M131=M131-1
4360 LET M131=M131+1
4370 PRINT - BAR
4380

```

3-D MOVER

I have been interested for some time in trying to get a program to RUN on the ZX81 which displayed a solid object rotating in space giving a three-dimensional effect.

The program, once typed in, requires a full five minutes RUNNING time to complete the POKEing of each of the six picture frames. The program will then automatically begin. The display is quite realistic — there is no screen flicker or jumping — and shows a rotating rectangle, spinning on a fixed axis. The rotational effect lasts for around 30 seconds and looks something like the illustration shown in figure 1.

SHAPES OF THINGS

The shape of the figure is easily altered by changing the co-ordinates, X, Y and Y1 in lines 140 to 800. (You'll need to do this on plotting paper though.)

One restriction of the program is that only the top half (lines 0 to 38) can be used. When I first developed the program, I tried to use the whole screen (but I ran out of memory given with the 16K RAM Pack attached!).

Once you have the program entered into the machine, it should be SAVED before you attempt to RUN it. Once it has been RUN, any new instructions

have to be typed in 'blind' as the PRINT position has been moved down below line 23 of the screen. If you want to re-RUN the program, you should type in GOTO 1000; this will also have to be done 'blind'.

Creating a three-dimensional image on your 16K ZX81 is not as difficult as you might imagine.

As a final point, the insertion of a 'E' sign at lines 630 and 660 is to end each frame with a code signal for the operating machine code to recognise that end of frame, and the end of the last (the sixth) frame.

1E FF LD E,255

Initialisation to repeat cycle 255 times; can be shortened by POKEing smaller numbers to address 30001.

1D 75 DBC E
FE 00 LD A,E
CB CP 0
3A 0C 40 RET IF ZERO
01 5A 75 LD HL,10000
LD BC,30042

Start of display.
Set address of first data bit of first frame.

0A LD A(BC)
77 LD(H,HL)
23 INC HL
03 INC BC
0A LD A(BC)
FE 0C CP 12
20 FB IR NZ 8 steps back

Prints it.
Next screen position.
Next data bit.

03 INC BC
0A LD A(BC)
FE 0C CP 12
28 E6 IR Z 30 steps back

Compare it to 'E'.
If not 'E', print it and continue with next data.
Next data bit.

16 02 LD D,02
15 DBC D
7A LD A,D
FE 00 CP 0
30 FA IR NZ 6 steps back
2A 0C 40 LD HL,10000
0A LD A(BC)
18 E4 IR 28 steps back

Compare it to 'E'.
If it is 'E', return to start and repeat the cycle. If not 'E', delay after printing first frame.
Delay — POKEing larger number than two at address 30009 will reduce the speed of rotation.

GO TO screen and start again.

The machine code part of the listing.

JUST ANOTHER BRICK

Did anyone ever say to you "Gee you're a real brick"? Did it make you see red? Well now give vent to your anger... Demolish walls and draw lines round your ZX81.

The wall is closing in on you. There's no escape. Can you destroy it before it destroys you? The answer, quite simply, is NO. The wall is endless. Your 'V' is moved rapidly from left to right at the top of the screen. The wall scrolls up towards you. Your only chance is to hurl yourself at the wall in the hope of dislodging some bricks.

The game is lost for a BASIC program. This is because characters are POKEd into the display file — much faster than PRINT AT, especially when dealing with numbers. Your score is POKEd on the bottom line of the screen, making use of the full 24 lines. It is necessary to use these lines as they are not moved up when SCROLL is used.

The speed of the wall moving can be made faster/slower by changing the length of the loop in line 250. You could even introduce another variable which makes the game faster as it goes along.

VARIABLES USED

P = Display file for poking the

'V'.

W = Display file for poking your score.

S = Score.

H = High Score.

X = General purpose loops

Y = Position for the 'V'.

C = Counter for number of bricks you have hit.

D = Number of bricks to be knocked out.

Z = Part of score being poked.

F = Vertical position for 'V' when being fired at the wall.

AS = String array for printing "SPLAT" in big letters.

X3 = Score string to be poked on line 24.

SURROUND

The idea of this game is to trap your opponent, the ZX81, by making it bump into something. The ZX81 will try and trap you, using a rare blend of cunning and intelligence.

Despite this, you'll probably find you know what its next move is going to be after playing it a few times. Then you can get to work, and modify the computer's strategy by changing lines 540 to 570.

The first player to win five games wins the round.

1178 NEXT X
1180 PRINT AT 9,1:"YOU HAVE BEEN
CRASHED TO PIECES BY THE U.S.S."

```

1400 FOR K=0 TO 31
1410 PRINT AT X,1;"

```

```

0010 IF ENKEY=1 THEN GOTO 0020
0020 CLS

```

Forward String



```

10 CLS
20 LET Z=0
30 LET O=0
40 LET B=0
50 LET N=255
70 LET P=PEEK 16396+PEEK 16397
8000 *1
80 LET B=324

130 FOR C=0 TO 99
135 PRINT AT C,0; "++"
140 FOR C
150 FOR Z=0 TO 31
160 PRINT AT B,C;" "
170 PRINT AT B,C;" "
180 NEXT C
190 LET C=1
195 PRINT AT B,3;"ZKX1-2:THE
21 "YOU" Y
210 LET B=B+C
220 IF PEEK (P+B) < 0 THEN GOSUB
230
250 LET B=B+8

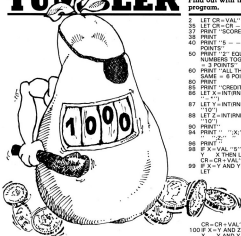
```

```

255 LET B=JUNKY
270 IF B=0 OR B=1 THEN LET
T B=0
300 LET B=0+(B=0)*1-(B=1)*1+1
310 LET B=1+(B=1)*1-(B=0)*1+1
320 IF PEEK (P+B) < 0 THEN GOTO
1000
330 POKE P+B,120
340 IF PEEK (P+B) < 0 THEN GOTO
1000
350 POKE P+B,120
360 GOTO 300
370 LET B=B+C
380 LET C=32
390 IF PEEK (B+P+1) < 0 THEN LET
C=1
400 IF PEEK (B+P+1) < 0 THEN LET
C=1
410 IF PEEK (B+P+32) < 0 THEN LET
C=32
420 LET B=B+C
430 RETURN
440 CLS
450 PRINT "THE ZKX1 WIND YET AG
AIN"
460 LET Z=Z+1
470 PRASE 464
480 IF Z=1 THEN GOTO 40
490 PRINT "ANOTHER ROUND TO ME"
500 GOTO 1600
510 CLS
520 PRINT "YOU WON, WHAT A BRKE
OCCASION"
530 LET Y=Y+1
540 PRASE 484
550 IF Y=1 THEN GOTO 40
560 PRINT "A ROUND TO YOU - BRH
2100"
570 PRINT "ZKX1-2:YOU=Y"
580 PRINT "ANOTHER ROUND IF ON
N"
590 LET B=JUNKY
600 IF B=0 THEN GOTO
610 IF B=1 THEN GOTO 1600

```


NUMBER TUMBLER



Written for the unexpanded Z801, this program is a version of the old favorite 'Fruit machine'. However, in this program, instead of designs, bars, apples, etc., you use numbers.

At the beginning of the game, you have 10 points in credit. These are displayed on a small score table on the screen. When you touch and key, the game starts and those numbers are displayed. If you get any two numbers the same beside each other, you are awarded

three points; if you get all three numbers the same you will be awarded the maximum of six points. Should you get the number 5 in the first position, you get a bonus score of two points.

The number of credit points you have decreases by one point for each turn. The game ends when your credit rating reaches zero. As there is limited memory left over, I was unable to include a 'hold' feature, but I'm sure that those of you with extra memory will be able to write a quick routine.

Is your number up?
Find out with this little program.

```

3  LET CR=VAL" 10"
35 LET CR=CR - VAL "1"
37 PRINT "SCORE TABLE"
38 PRINT
40 PRINT "5 - - - 2
   POINTS"
50 PRINT "2" EQUAL
   NUMBERS TOGETHER
   = 3 POINTS"
60 PRINT "ALL THREE THE
   SAME = 6 POINTS"
60 PRINT
85 PRINT "CREDITS = "CR
85 LET X=INT(RND * VAL
   "10")
87 LET Y=INT(RND * VAL
   "10")
88 LET Z=INT(RND * VAL
   "10")
90 PRINT
94 PRINT "X:" "Y:"
   "Z:"
96 PRINT
98 IF X=VAL "5" AND
   Y = X THEN LET
   CR=CR+VAL"2"
99 IF X=Y AND Y=Z THEN
   LET

```

```

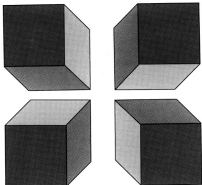
CR=CR+VAL"6"
100 IF X=Y AND Z = X OR
   X = Y AND Y=Z THEN
   LET CR=CR+VAL"3"
105 IF CR=PI - PI THEN
   GOTO VAL"130"
107 PRINT "TOUCH ANY
   KEY"
110 PAUSE 454
115 CLS
120 GOTO VAL "35"
130 PRINT "FINISHED - NO
   CREDITS LEFT"

```

Note that the message 'Score TABLE' in line 37 should be in inverse video for maximum effect.

3D CUBES

Here is a way to get representations of 3-D cubes on the ZX81. The method is explained and uses a 'selfmade television screen worksheet'.



Here's the way my program works. All you need is a 16K ZX81, a worksheet like the one I've shown here, and a little patience. Although it takes a while to type the program into your ZX81, the results seem pretty good to me.

HOW IT RUNS

Lines 5 to 69 ask for the four co-ordinates of the angular points of the upper surface of the cube. You only have to input the first co-ordinate of the first angular point (E,F), then press NEWLINE (of course), and then input the second co-ordinate, i.e. F Mind the right

order of the angular points! Look first of all the television screen worksheet to know where (E,F), (G,H), (I,J), (K,L) are situated, and don't change the order.

Line 70 asks you to give the height of the cube. Don't make M too high otherwise the cube will 'run out of the screen'. A height of about 10 to 25 is recommended. Of course, it all depends upon the position of the chosen co-ordinates of the angular points.

Lines 85 to 180 print the first 'horizontal' edge (see the second example) of the upper surface of the cube. In the first

example the first line of the upper surface at the left will be printed.

Lines 185 to 280 print the second 'horizontal' edge.

Lines 285 to 340 print the four vertical edges of the cube.

Lines 345 to 440 will print the first horizontal edge of the base (see example two) or the first line in example one.

Lines 445 to 540 do the same as lines 345 to 440. Then the second line will be printed.

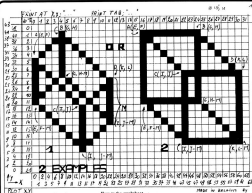
Lines 545 to 700 print the four oblique lines of the cube.

As you can see the program is rather easy to understand because it's all BASIC.

```

1 REM ***3-DIMENSIONAL CURVES***
2 GOSUB 10000 - RELAJOUR**
3 INPUT "INPUT ANGULAR POINTS OF THE DESIRED SURFACE"
4 PRINT
5 PRINT "INPUT ANGULAR POINT"
6 R=C,F
7 INPUT E
8 INPUT F
9 PRINT "INPUT ANGULAR POINT"
10 R=C,H
11 INPUT G
12 INPUT H
13 PRINT "INPUT ANGULAR POINT"
14 C=I,J
15 INPUT I
16 INPUT J
17 PRINT "INPUT ANGULAR POINT"
18 D=K,L
19 INPUT K
20 INPUT L
21 PRINT "WHAT IS THE RIGHT OF THE CURVE?"
22 CLS
23 IF F=H THEN GOTO 95
24 IF F=H THEN GOTO 115
25 FOR H=0 TO 5
26 PLOT H,F
27 NEXT H
28 GOTO 135
29 LET Q=ABS (F-H)
30 FOR P=0 TO Q-1
31 IF E=0 AND F=H THEN GOTO 15
32 IF E=0 AND F=H THEN GOTO 14
33 IF E=0 AND F=H THEN GOTO 16
34 IF E=0 AND F=H THEN GOTO 17
35 PLOT E=F,F=F
36 GOTO 185
37 PLOT E=F,F=F
38 GOTO 195
39 PLOT E=F,F=F
40 GOTO 205
41 PLOT E=F,F=F
42 GOTO 215
43 PLOT E=F,F=F
44 GOTO 225
45 IF J=L THEN GOTO 105
46 IF J=L THEN GOTO 215
47 FOR Q=1 TO K
48 PLOT Q,J
49 NEXT Q
50 GOTO 235
51 LET R=ABS (J-L)
52 FOR S=0 TO R-1
53 IF I=K AND J=L THEN GOTO 25
54 IF I=K AND J=L THEN GOTO 24
55 IF I=K AND J=L THEN GOTO 26
56 IF I=K AND J=L THEN GOTO 27
57 PLOT I=S,J=S
58 GOTO 285
59 PLOT I=S,J=S
60 GOTO 295
61 PLOT I=S,J=S
62 GOTO 305
63 PLOT I=S,J=S
64 GOTO 315
65 NEXT S
66 FOR T=J-H TO J
67 PLOT T,T
68 NEXT T
69 FOR U=L-H TO L
70 PLOT U,U
71 NEXT U
72 FOR V=F-H TO F
73 PLOT V,V
74 NEXT V
75 FOR W=H-H TO H
76 PLOT W,W
77 NEXT W
78 IF J=H=L-H THEN GOTO 355
79 IF J=H=L-H THEN GOTO 375
80 FOR X=1 TO K
81 PLOT X,J-H
82 NEXT X
83 LET Y=ABS (J-L)
84 FOR Z=0 TO Y-1
85 IF I=K AND J=H=L-H THEN GOTO 415
86 IF I=K AND J=H=L-H THEN GOTO 435
87 IF I=K AND J=H=L-H THEN GOTO 455
88 IF I=K AND J=H=L-H THEN GOTO 475
89 PLOT X,Z
90 NEXT X
91 LET Y=ABS (F-H)
92 FOR Z=0 TO Y-1
93 IF E=0 AND F=H=H THEN GOTO 485
94 IF E=0 AND F=H=H THEN GOTO 495
95 IF E=0 AND F=H=H THEN GOTO 505
96 IF E=0 AND F=H=H THEN GOTO 515
97 PLOT E=0,F=H=0
98 GOTO 525
99 PLOT E=0,F=H=0
100 NEXT Q
101 LET Q=ABS (H-L)
102 FOR S=0 TO Q-1
103 IF E=I AND F=H=J-H THEN GOTO 535
104 IF E=I AND F=H=J-H THEN GOTO 555
105 PLOT I=0,J=H=0
106 GOTO 565
107 PLOT I=0,J=H=0
108 NEXT Q
109 LET Q=ABS (H-L)
110 FOR S=0 TO Q-1
111 IF E=I AND F=J=J-H THEN GOTO 545
112 IF E=I AND F=J=J-H THEN GOTO 565
113 PLOT I=S,J=S
114 GOTO 575
115 PLOT I=S,J=S
116 NEXT S
117 LET T=ABS (H-L)
118 FOR U=0 TO T-1
119 IF G=K AND H=L THEN GOTO 585
120 IF G=K AND H=L THEN GOTO 595
121 PLOT G=U,L=U
122 GOTO 605
123 PLOT G=U,L=U
124 NEXT U

```



Four of the best on BBC

PROGRAMMING THE ERC MICRO

Full Text: [View](#) | [Print](#) | [Email](#) | [Link](#)

²For parallel reviews of the potential of the 800 Micro... www.800micro.com

To help you program your BBC Micro, Practical programming points and BBC BASIC as applied to graphics, words, numbers and sound are covered as well as machine code, hex, assembly, pointers, introduction, file handling and file systems.

Table 1

1000

6502 MACHINE CODE FOR BEGINNERS

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

SQL*Plus is a simple language to learn, but it takes up a lot of memory space and is slow to run. Machine code is more complex, but it does have considerable benefits and this book shows you how to take advantage of them.

[illegible]

1000

EXPLOITING BNC BASIC

A. P. Söderström and B. J. Söderström

A book for BBC Micro users who have mastered the first stages of programming and who now want to exploit their machine to its

Two transactions also available containing some of the largest
 assets.

[illegible]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----



BBC MICRO PROGRAMS IN BASIC

Keywords: *depression, anxiety, self-esteem, self-efficacy, self-esteem, self-efficacy*

Twenty-seven games programs, each with an explanation of how it works and how to play it. They're not only fun, they'll improve your BASIC programming skills at the same time.

TABLE 1

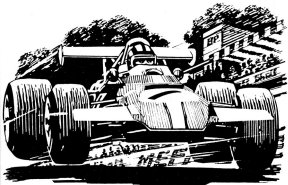
TABLE 1

On and on our copies move

Newnes Technical Books, Borough Green, Kent TN11 9SP

GRAND PRIX DRIVER

Strap yourself in and
prepare yourself for the
race of your life!



This program, occupying almost 15K, simulates a 10 lap race around a Grand Prix circuit involving all the thrills and spills of the real thing!

Racing around the track clockwise, you move your car using the 'S' key to move to the left, the 'R' key to move right, the 'F' key to move up and the 'D' key to move down.

A GRAND TIME

You'll have to watch out for various obstacles on the track, because if you hit them they will cause you to incur time penalties. If you accidentally hit the outer barrier of the track, your car will crash and you are out of the race instantly. Also, if you have an accident in the pits, it is supposed your car will

be blown to pieces. Should you crash into the inner barrier, your car will automatically be towed to the pits where it will undergo extensive repairs which will cause you to lose a lap. However, if you manage to crash your car into the barrier again, your car will burst into flames and you are placed out of the race for good!

You begin the race with a

GRAND PRIX DRIVER

full tank of petrol — 800 litres — which should be enough for about seven to eight laps if you drive carefully. After you have completed seven laps you may enter the pits and be re-fuelled with 50 litres of petrol. You can drive through the pits before this stage of the race, but you will not receive any extra fuel.

When you drive into the pits you must take care as there is usually a lot of oil carelessly strewn around which makes skidding very easy to do.

NO CHEATING

*Various routines have been put in making it more hard for you.

to cheat in this game. You are not allowed to drive through the pits backwards (that re-teeling move is only once a lap), and you may not drive backwards and forwards across the start/finish line.

Various indicators, such as fuel remaining* and lap number*, are displayed and updated each time you cross the start/finish line (or when you crash) giving you some idea of your progress throughout the course.

[illegible]

```

EMERGED. USUALLY BY LOSS OF TI
R. HOWEVER, HITTING THE DUTCH
C. SADDLES. PUT YOU OUT OF
S. RACE IMMEDIATELY AND ON GROUN
NT IN THE PITS WILLBLAM YOU TO C
NITHERERS.
430 PRINT AT 0.0: "COLLIDING WIT
4 THE JAGGED CRASH BARRIER CHASE
S. PRINCE OF THE LIVE CHARGE. RESCU
KING A PIT STOP - YOUR CAR WILL
SC TOWED THERE - AUTOMATICALLY
AND YOU WILL LOSE ONE COMPLETE
LAP."
440 PRINT AT 10.0: "HITTING THE
INNER CRASH BARRIER DEPOSE YOU H
AVE COMPLETED 1 FULL LAP MORE
THAN ONCE! WILL SET YOU ON FIRE
ON FIRE AND PUT YOU OUT OF THE RACE
PERMANENTLY."
450 PRINT AT 11.0: "PRESS ANY KE
Y TO CONTINUE"
470 PRINCE 404
480 CLS
490 PRINT AT 1.0: "YOU START THE
RACE WITH A FULL TANK OF FUEL."
500 PRINT AT 2.0: "AFTER COMPLET
ING 2 FULL LAPS YOU MAY REFUEL
E SIMPLY BY DRIVING INTO THE PITS
. BEFORE THIS, YOU CAN DRIVE THE
CASH THE PITS IF YOU WISH, BUT
WILL RECEIVE NO ADDITIONAL FU
EL."
510 PRINT AT 10.0: "YOU WILL REC
EIVE ONLY AN ADDIT- TIONAL 50 LIT
RES ON EACH REFUELLING STOP."
520 PRINT AT 11.0: "WAIT UNTIL A
SUELLING HAS BEEN COMPLETED. BE
FORE ENTERING THE PITS ON."
570 PRINT AT 10.0: "DRIVING ON A
ROAD ON THE PIT ROAD, WHICH
IS COVERED IN OIL."
580 PRINT AT 11.0: "PRESS ANY KE
Y TO CONTINUE"
590 PRINCE 404
600 CLS
610 PRINT AT 0.0: "THERE ARE OTH
ER HAZARDS AND THE ROAD SURFACE
IS NOW SLIPPERY. HAZARDS - YOU
WILL HAVE TO DRIVE CAREFULLY."
620 PRINT AT 0.0: "YOU ARE NOT A
LLOWED TO DRIVE BACKWARDS. THE
DUTCH Y. DRIVE THROUGH THE C
TANT A FINISH LINE."
610 PRINT AT 0.0: "ON LEAVING TH
E PITS, YOU MAY NOTIFY DRIVEN BY
THE SCREEN AND YOU CANST Y
DEACONS. BEWARE. MORE THAN INC
E ON THE DRIVE LIP."
630 PRINT AT 10.0: "THE "FUEL G
EATING" AND LAP NO" DID
LOVE WILL ONLY GASTE AS YOU C
ASS THE START A FINISH POINT O
ISPLAYED IN UNITS OF - IF YOU S
OULD MAKE AN ACCIDENT - IF YOU S
620 PRINT AT 10.0: "HOW PRESS AN
Y KEY TO PLAY."
640 PRINT AT 11.0: "**** HAPPY C
RUIED *****"
700 PRINCE 404
710 GOTO 370
000 LET Z=10
010 LET A=10
020 LET X=11
030 IF A=0 AND Z=24 THEN LET A=
A+1
040 IF A=0 AND Z=25 THEN LET A=
A+1
050 IF A=10 AND Z=13 THEN LET A
=1
060 IF A=1 AND Z=26 THEN GOSUB
0800
070 IF A=1 AND Z=27 THEN LET Z=
Z+1
010 IF A=14 AND Z=13 THEN GOTO
2000
090 IF INKEY$="S" THEN LET Z=Z-
1
060 IF INKEY$="7" THEN LET A=0-
1
070 IF INKEY$="8" THEN LET A=0-
1
080 IF INKEY$="8" THEN LET Z=Z+
1
090 PRINT AT A.Z: "AT A.Z: "
1000 IF A=0 THEN GOTO 1000
1010 IF A=10 THEN GOTO 1000
1020 IF Z=0 THEN GOTO 1000
1030 IF Z=1 THEN GOTO 1000
1040 IF Z=2 AND Z=3 AND Z=20 TH
EN GOTO 1000
1050 IF A=10 AND Z=13 AND Z=13 T
HEN GOSUB 1000
1060 IF A=14 AND Z=14 THEN GOSUB
1000
1070 IF Z=0 AND Z=0 AND Z=7 THEN
GOSUB 1000
1080 IF A=11 AND Z=0 THEN GOSUB
1000
1090 IF A=4 AND Z=25 THEN GOSUB
1000
1100 IF A=10 AND Z=25 THEN GOSUB
1000
1110 IF A=7 AND Z=25 AND Z=25 TH
EN GOSUB 1000
1120 IF Z=10 AND A=5 THEN GOSUB
1000
1130 IF Z=23 AND A=0 AND A=7 THE
N GOSUB 1000
1140 IF A=17 AND Z=0 THEN LET Z=
0
1150 IF A=10 AND Z=0 THEN LET Z=
0
1160 IF A=0 AND Z=1 THEN LET A=0
1170 IF A=0 AND Z=2 THEN LET A=0
1180 IF A=0 AND Z=3 THEN LET A=0
1190 IF A=0 AND Z=11 THEN LET A=
0
1200 IF A=0 AND Z=21 THEN LET A=
0
1210 IF A=0 AND Z=10 THEN LET Z=
0
1220 IF A=0 AND Z=20 THEN LET Z=
0
1230 IF A=14 AND Z=25 THEN LET A
=0
1240 IF A=15 AND Z=27 THEN LET A
=0
1250 IF A=10 AND Z=20 THEN LET A
=0
1260 IF A=10 AND Z=17 THEN LET A
=0
1270 IF Y=000 THEN GOTO 000
FOR Y=0 TO 10
1280 PRINT AT 0.0: "*****"
1290 PRINT AT 0.0: "*****"
1300 NEXT Y
1310 PRINCE 400
1320 PRINT AT 10.0: "*****"
1330 PRINT AT 11.0: "*****"
1340 GOTO 1010
1350 FOR L=0 TO 10
1360 PRINT AT 11.0: "*****"
1370 PRINT AT 11.10: "UNOONPH"
1380 NEXT L
1390 PRINCE 100
1400 PRINT AT 11.0: "*****"
1410 PRINT AT 7.0: "*****"
1420 PRINT AT 0.0: "*****"
1430 PRINT AT 0.0: "*****"
1440 GOTO 1100
1450 FOR J=0 TO 10
1460 PRINT AT 11.0: "*****"
1470 PRINT AT 11.10: "CRASH"
1480 NEXT J
1490 PRINCE 100
1500 PRINT AT 11.0: "*****"
1510 PRINT AT 0.0: "*****"
1520 PRINT AT 0.0: "*****"
1530 PRINT AT 10.0: "*****"
1540 PRINT AT 11.0: "*****"
1550 PRINCE 200

```

```

1028 CLS
1029 IF S=2 THEN GOTO 2000
1030 FOR S=0 TO 10
1031 PRINT AT S,2: "-"
1032 PRINT AT S,0: "-"
1033 PRINT AT S,0: "-"
1034 PRINT AT S,0: "THE CURRENT R
ACE LEADER IS"
1035 PRINT AT S,0,00: "I 'J.E.' LA
NCE"
1036 NEXT S
1037 PRASE 100
1038 CLS
1039 PRINT AT 14,0: "ANOTHER GAME"
1040 INPUT R$
1041 IF R$="YES" THEN GOTO 210
1042 IF R$="NO" THEN GOTO 1000
1043 PRINT AT 14,0: "U BET ONLY A
HUMAN"
1044 IF R$="YES" OR "NO"
1045 PRASE 100
1046 GOTO 1000
1047 CLS
1048 PRINT AT 10,2: "OK. HOPE YOU
ENJOYED PLAYING"
1049 PRINT AT 14,7: "GRAND PRIX
DRIVER"
1050 PRINT AT 10,0: "BYE BYE FOR
NOW"
1051 STOP
1052 LET S=0-1
1053 IF S=0 THEN GOTO 1000
1054 FOR S=0 TO 10
1055 PRINT AT 11,0: "-"
1056 PRINT AT 11,10: "CRUNCH"
1057 NEXT S
1058 PRASE 100
1059 PRINT AT 11,0: "-"
1060 PRINT AT 7,0: "-"
1061 PRINT AT 0,0: "-"
1062 PRINT AT 0,7: "-"
1063 PRINT AT 10,0: "-"
1064 PRASE 100
1065 PRINT AT 11,7: "-"
1066 PRINT AT 10,10: "-"
1067 PRASE 100
1068 PRINT AT 11,0: "-"
1069 PRINT AT 7,0: "-"
1070 PRINT AT 0,0: "-"
1071 PRINT AT 0,0: "-"
1072 LET Z=0
1073 LET S=0
1074 LET X=0-10
1075 PRASE 000
1076 SOUND 1000
1077 PRINT AT A,2: "AT A,2: "
1078 RETURN
1079 FOR S=0 TO 10
1080 PRINT AT 11,0: "-"
1081 PRINT AT 11,11: "SSSSSSSSSS"
1082 NEXT S
1083 PRINT AT 11,0: "-"
1084 PRINT AT 7,0: "-"
1085 PRINT AT 0,0: "-"
1086 PRASE 100
1087 PRINT AT 11,7: "-"
1088 PRINT AT 10,0: "-"
1089 LET Z=0
1090 LET S=0
1091 LET X=X+10
1092 PRASE 000
1093 SOUND 1000
1094 PRINT AT A,2: "AT A,2: "
1095 RETURN
1096 FOR U=0 TO 10
1097 PRINT AT 11,0: "-"
1098 PRINT AT 11,10: "SHOOOSH"
1099 NEXT U
1100 PRASE 100
1101 PRINT AT 11,0: "-"
1102 PRINT AT 7,0: "-"
1103 PRINT AT 0,0: "-"
1104 PRINT AT 0,0: "-"
1105 PRASE 000
1106 PRINT AT 11,7: "-"
1107 PRINT AT 10,0: "-"
1108 PRASE 000
1109 LET Z=X+50
1110 LET S=0

```


[illegible]

```

0768 PRINT AT 9.13: " "
0769 PRINT AT 10.0: " "
0770 PRINT AT 14.10: " "
0771 PRINT AT 17.0: " "
0772 PRINT AT 18.0: " "
0773 PRINT AT 19.0: " "
0774 PRINT AT 9.11: " "
0775 PRINT AT 9.4: " "
0776 PRINT AT 7.9: " "
0777 POINT AT 9.13: " "
0778 POINT AT 10.0: " "
0779 PRINT AT 14.10: " "
0780 PRINT AT 17.0: "THE DRIVER I"
0781 GOTO PLACE
0782 PRINT AT 10.0: "IS NOW "
0783 PRINT AT 10.11: "ON "
0784 NEXT P
0785 PHASE 3000
0786 GOTO 1000
0787 LET S=5+1
0788 GOINT AT 9.11: " "
0789 GOINT AT 9.11: "000-X"
0790 GOINT AT 11.10: " "
0791 IF S=10 THEN GOTO 2000
0792 LET R=R+1
0793 GOTO 2000
0794 LET S=S-1
0795 LET F=F-1
0796 LET DE="NO ONE"
0797 RETURN
0798 IF S=1 THEN GOSUB 3000
0799 LET S=S+1
0800 PRINT AT 0.2: "P": AT 0.2: " "
0801 RETURN
0802 LET X=X-50
0803 FOR H=1 TO 20
0804 GOINT AT 9.11: " "
0805 PRINT AT 1.20: "FUELING"
0806 NEXT H
0807 GOINT AT 9.11: "000-X"
0808 GOTO 1000
0809 PRINT AT 0.0: " "
0810 PRINT AT 0.0: " "
0811 PRINT AT 0.0: " "
0812 PRINT AT 0.0: " "
0813 PRINT AT 0.0: " "
0814 PRINT AT 0.0: " "
0815 PRINT AT 0.0: " "
0816 PRINT AT 0.0: " "
0817 PRINT AT 0.0: " "
0818 PRINT AT 0.0: " "
0819 PRINT AT 0.0: " "
0820 PRINT AT 0.0: " "
0821 PRINT AT 0.0: " "
0822 PRINT AT 0.0: " "
0823 PRINT AT 0.0: " "
0824 PRINT AT 0.0: " "
0825 PRINT AT 0.0: " "
0826 PRINT AT 0.0: " "
0827 PRINT AT 0.0: " "
0828 PRINT AT 0.0: " "
0829 PRINT AT 0.0: " "
0830 PRINT AT 0.0: " "
0831 PRINT AT 0.0: " "
0832 PRINT AT 0.0: " "
0833 PRINT AT 0.0: " "
0834 PRINT AT 0.0: " "
0835 PRINT AT 0.0: " "
0836 PRINT AT 0.0: " "
0837 PRINT AT 0.0: " "
0838 PRINT AT 0.0: " "
0839 PRINT AT 0.0: " "
0840 PRINT AT 0.0: " "
0841 PRINT AT 0.0: " "
0842 PRINT AT 0.0: " "
0843 PRINT AT 0.0: " "
0844 PRINT AT 0.0: " "
0845 PRINT AT 0.0: " "
0846 PRINT AT 0.0: " "
0847 PRINT AT 0.0: " "
0848 PRINT AT 0.0: " "
0849 PRINT AT 0.0: " "
0850 PRINT AT 0.0: " "
0851 PRINT AT 0.0: " "
0852 PRINT AT 0.0: " "
0853 PRINT AT 0.0: " "
0854 PRINT AT 0.0: " "
0855 PRINT AT 0.0: " "
0856 PRINT AT 0.0: " "
0857 PRINT AT 0.0: " "
0858 PRINT AT 0.0: " "
0859 PRINT AT 0.0: " "
0860 PRINT AT 0.0: " "
0861 PRINT AT 0.0: " "
0862 PRINT AT 0.0: " "
0863 PRINT AT 0.0: " "
0864 PRINT AT 0.0: " "
0865 PRINT AT 0.0: " "
0866 PRINT AT 0.0: " "
0867 PRINT AT 0.0: " "
0868 PRINT AT 0.0: " "
0869 PRINT AT 0.0: " "
0870 PRINT AT 0.0: " "
0871 PRINT AT 0.0: " "
0872 PRINT AT 0.0: " "
0873 PRINT AT 0.0: " "
0874 PRINT AT 0.0: " "
0875 PRINT AT 0.0: " "
0876 PRINT AT 0.0: " "
0877 PRINT AT 0.0: " "
0878 PRINT AT 0.0: " "
0879 PRINT AT 0.0: " "
0880 PRINT AT 0.0: " "
0881 PRINT AT 0.0: " "
0882 PRINT AT 0.0: " "
0883 PRINT AT 0.0: " "
0884 PRINT AT 0.0: " "
0885 PRINT AT 0.0: " "
0886 PRINT AT 0.0: " "
0887 PRINT AT 0.0: " "
0888 PRINT AT 0.0: " "
0889 PRINT AT 0.0: " "
0890 PRINT AT 0.0: " "
0891 PRINT AT 0.0: " "
0892 PRINT AT 0.0: " "
0893 PRINT AT 0.0: " "
0894 PRINT AT 0.0: " "
0895 PRINT AT 0.0: " "
0896 PRINT AT 0.0: " "
0897 PRINT AT 0.0: " "
0898 PRINT AT 0.0: " "
0899 PRINT AT 0.0: " "
0900 PRINT AT 0.0: " "
0901 PRINT AT 0.0: " "
0902 PRINT AT 0.0: " "
0903 PRINT AT 0.0: " "
0904 PRINT AT 0.0: " "
0905 PRINT AT 0.0: " "
0906 PRINT AT 0.0: " "
0907 PRINT AT 0.0: " "
0908 PRINT AT 0.0: " "
0909 PRINT AT 0.0: " "
0910 PRINT AT 0.0: " "
0911 PRINT AT 0.0: " "
0912 PRINT AT 0.0: " "
0913 PRINT AT 0.0: " "
0914 PRINT AT 0.0: " "
0915 PRINT AT 0.0: " "
0916 PRINT AT 0.0: " "
0917 PRINT AT 0.0: " "
0918 PRINT AT 0.0: " "
0919 PRINT AT 0.0: " "
0920 PRINT AT 0.0: " "
0921 PRINT AT 0.0: " "
0922 PRINT AT 0.0: " "
0923 PRINT AT 0.0: " "
0924 PRINT AT 0.0: " "
0925 PRINT AT 0.0: " "
0926 PRINT AT 0.0: " "
0927 PRINT AT 0.0: " "
0928 PRINT AT 0.0: " "
0929 PRINT AT 0.0: " "
0930 PRINT AT 0.0: " "
0931 PRINT AT 0.0: " "
0932 PRINT AT 0.0: " "
0933 PRINT AT 0.0: " "
0934 PRINT AT 0.0: " "
0935 PRINT AT 0.0: " "
0936 PRINT AT 0.0: " "
0937 PRINT AT 0.0: " "
0938 PRINT AT 0.0: " "
0939 PRINT AT 0.0: " "
0940 PRINT AT 0.0: " "
0941 PRINT AT 0.0: " "
0942 PRINT AT 0.0: " "
0943 PRINT AT 0.0: " "
0944 PRINT AT 0.0: " "
0945 PRINT AT 0.0: " "
0946 PRINT AT 0.0: " "
0947 PRINT AT 0.0: " "
0948 PRINT AT 0.0: " "
0949 PRINT AT 0.0: " "
0950 PRINT AT 0.0: " "
0951 PRINT AT 0.0: " "
0952 PRINT AT 0.0: " "
0953 PRINT AT 0.0: " "
0954 PRINT AT 0.0: " "
0955 PRINT AT 0.0: " "
0956 PRINT AT 0.0: " "
0957 PRINT AT 0.0: " "
0958 PRINT AT 0.0: " "
0959 PRINT AT 0.0: " "
0960 PRINT AT 0.0: " "
0961 PRINT AT 0.0: " "
0962 PRINT AT 0.0: " "
0963 PRINT AT 0.0: " "
0964 PRINT AT 0.0: " "
0965 PRINT AT 0.0: " "
0966 PRINT AT 0.0: " "
0967 PRINT AT 0.0: " "
0968 PRINT AT 0.0: " "
0969 PRINT AT 0.0: " "
0970 PRINT AT 0.0: " "
0971 PRINT AT 0.0: " "
0972 PRINT AT 0.0: " "
0973 PRINT AT 0.0: " "
0974 PRINT AT 0.0: " "
0975 PRINT AT 0.0: " "
0976 PRINT AT 0.0: " "
0977 PRINT AT 0.0: " "
0978 PRINT AT 0.0: " "
0979 PRINT AT 0.0: " "
0980 PRINT AT 0.0: " "
0981 PRINT AT 0.0: " "
0982 PRINT AT 0.0: " "
0983 PRINT AT 0.0: " "
0984 PRINT AT 0.0: " "
0985 PRINT AT 0.0: " "
0986 PRINT AT 0.0: " "
0987 PRINT AT 0.0: " "
0988 PRINT AT 0.0: " "
0989 PRINT AT 0.0: " "
0990 PRINT AT 0.0: " "
0991 PRINT AT 0.0: " "
0992 PRINT AT 0.0: " "
0993 PRINT AT 0.0: " "
0994 PRINT AT 0.0: " "
0995 PRINT AT 0.0: " "
0996 PRINT AT 0.0: " "
0997 PRINT AT 0.0: " "
0998 PRINT AT 0.0: " "
0999 PRINT AT 0.0: " "

```

EQUATION SOLVER

Suppose that you want to solve an equation of the form $K(x) = 0$, such as:

$$y = f(x) = x^3 + 3x^2 - 10x - 15$$

The roots of this equation are the values of x for which $y = 0$. You can find roughly where they occur from a sketch graph (Fig. 1) which shows that the curve cuts the x axis at three places. One is between $x = -5$ and $x = -4$, another between $x = -1$ and $x = 0$, and the third is between $x = 2$ and $x = 3$.

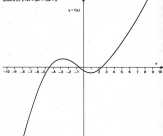
You could find this same information by tabulating the values of $y = f(x)$ for a range of values between, say, -10 and $+10$ (Fig. 2). Where the sign changes from plus to minus, or vice versa, is the interval in which you could expect to find a root. The table shows that there are again the three intervals $(-5, -4)$, $(-1, 0)$ and $(2, 3)$ which need to be studied in more detail.

The bisection method takes each interval in turn, divides it into two equal parts, and decides in which half the root is to be found. It then divides that half into two equal parts and repeats the process. This iteration continues until the size of the portion so formed is smaller than the specified error, at which point it gives the mid-point of the portion as the root of the equation. It is here that the graph crosses the x axis and changes sign.

ADVANTAGES AND DISADVANTAGES

Unless you tabulate the function over a very wide range, you cannot be sure of finding all the roots of an equation. Nor can you find any roots that might be complex numbers, such as the solution to the equation $y =$

GRAPH OF $y = x^3 + 3x^2 - 10x - 15$



$x^2 + 1$, for its graph does not even cross the x axis (Fig. 3). However, you can find the solutions to whatever accuracy you like given sufficient iterations and, as the method does not suffer from instability, it always converges to the required answer.

BETWEEN THE LINES

The program will run with about 48K RAM. The following describes the structure of the program.

Lines 10-110 set up the default values of the parameters, the equation to be solved is held as A5.

Lines 200-360 print the menu and choose the selected subroutine.

Lines 1000-1030 allow the equation to be entered or changed. To prevent program execution stopping with the report code A, powers of x such

We show you how to solve equations on the ZX81 using the bisection method.

as x^2 should be entered as $x \cdot x \cdot x$, and so on.

Lines 2000-2080 specify the interval in which you are searching for a root.

Lines 3000-3040 ask for the number values that you want to tabulate.

Lines 4000-4090 tabulate the function over the range set by option 2. If a change of sign occurs between the lowest and highest points of the range, the bisection method can be used.

If two or more changes of sign occur, the interval set by option 2 should be reduced to include just one such sign change. If no change of sign occurs, then a wider interval could be tried.

Lines 5000-5040 allow the maximum permitted error in the answer to be set. The smaller this is made, the longer it will take and the more iterations will be needed to reach the solution. In fact the error depends on $1/2^n$, where n is the number of

iterations.

Lines 6000-6200 carry out the actual bisection process, letting the user know if an incorrect interval has been set by option 2.

Lines 7000-7050 list the current values of all parameters, including the equation itself.

Lines 9000-9030 wait for a user to specify either a return to the menu or a printout of screen contents. The program will run as if it is even if no printer is available.

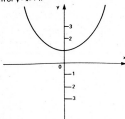
HOW TO USE THE PROGRAM

When first run, choose option 1 and enter test equation $x^2 + 3x - 10 = 5$ in the form:

$$X=X+X+3\cdot X-X-10\cdot X-5$$

Then choose option 2 and specify the interval from -10 to +10. Option 3 should be used to ask for 20 tabulation points. Option 4 will list the values of the equation at these 20 points, and will also tell you what to do when the screen fills up. The table should look like that in Fig. 1. Three changes of sign are found, so we call up option

GRAPH OF $y = x^2 + 1$.



2 again, and enter the interval $(-5, -4)$.

Next, we choose an error of less than 0.00001 using option 5. Option 6 calculates the root in our chosen interval as -4.84246 . You can then return to option 2 to set the other two intervals in which there is a root of the equation.

Note that the test equation is

a cubic, for which there can only be three roots. However, any polynomial or trigonometric equation can be entered, for example, $\sin(x) + \tan(x) - 3x^2 + 4x$.

With practice, this becomes an effective and reliable way of solving all sorts of equations — especially recommended for maths homework!

TABLE OF VALUES BETWEEN -10 AND 10 OF $X\cdot X+X+3\cdot X-X-10\cdot X-5$

X=-10	F(X)=-805
X=-9	F(X)=-481
X=-8	F(X)=-245
X=-7	F(X)=-131
X=-6	F(X)=-53
X=-5	F(X)=-5
X=-4	F(X)=19
X=-3	F(X)=25
X=-2	F(X)=19
X=-1	F(X)=7
X=0	F(X)=-5
X=1	F(X)=-11
X=2	F(X)=-5
X=3	F(X)=19
X=4	F(X)=67
X=5	F(X)=145
X=6	F(X)=259
X=7	F(X)=415
X=8	F(X)=619
X=9	F(X)=877
X=10	F(X)=1195

Fig.2.

MINEFIELD

A If you have to do in this game for your XM81 is to establish a safe route through a minefield you know to be planted with 13 deadly mines.

TO THE DEATH

Your mission is of the suicide

variety and of the utmost importance — vital to the war effort in the war you have happened upon. And that is why to a certain degree your lives are expendable. You command three men and their task is to beat a path through the minefield — at least one of your men must get through as

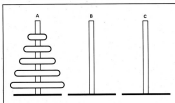
You must really watch your step in this program!

three explosions will alert the enemy that someone is trying to cross their defenses and your mission will be over.

You move your men across using the cursor keys. No clues or hints are given as to where the mines have been hidden — you must maneuver your men purely on instinct. Good luck!



TOWERS OF HANOI



This program will help you solve the Towers of Hanoi problem. This involves a number of rings of different sizes which are stacked on Tower 1 in a similar way to that shown in Fig. 1. The idea of the problem is to eventually stack the rings on Tower 2 or Tower 3 so that at no time does a larger ring rest on a smaller one.

A BIT STEEP?

The driving force of the program is the recursive subroutine (one which calls itself) at lines 90 to 104. The program enters the subroutine with N (the number of rings in play) and breaks out at line 82 with P (the tower number — 1, 2 or 3) from which the ring is to move, Q (the number of the 'goto' tower) and N (one less than the number of the moving ring — ring 1 being the smallest). After display of the move, the routine is re-entered at line 84 for the next instruction, and then circulates between lines 70 to 104 until the problem is solved. The program finally RETURNS to line 60 to

show the number of moves made (always $2^N - 1$, where N is the number of rings) and to wait for key 'P' at line 64.

The recursive subroutine can be modified to run on its own. First, you should add the following line:

```
40 INPUT N
```

where N represents the number of rings. Then, replace lines 60 to 68 with:

```
60 STOP
```

Finally, replace lines 82 to 84 with the following line:

```
82 PRINT "RING";  
N+1;"FROM TOWER";P;" TO  
TOWER"; Q
```

Then, RUN 40 and enter your value for N.

The action of the routine can be followed through, for N=3 for example, if it is remembered that GOSUB puts its RETURN address (the following line) onto the GOSUB stack and each RETURN removes an address (last in — first out) and goes to it.

The ZX81 solution to this age-old puzzler.

START HERE

For newcomers to the ZX81, the following notes may prove of interest.

Lines 40 to 46 produce a number of nine character rings, all PRINTed at column zero; their middle characters form the centre of Tower 1 at column four. The tops of the towers (usually invisible) are all on screen line five.

At lines 108 to 124, the P and Q values (1, 2 or 3) from line 82 are converted to D, 9 or 18, and are assigned to GF and GT as the PRINT AT column numbers for the 'go from' and 'goto' towers respectively.

The character, R\$(N+1), of the ring to be moved is given by N from line 82, but its line number and the height to which it must be raised are found by PEEKing the towers. This requires the display file reference address, D, which is the central top of Tower 1. (Try adding a line 49 POKE D,23.) The D file addresses of the tops of Towers 2 and 3 are therefore D + 9 and D + 18, so a ring at three lines below the top of Tower 2 will have its centre at addresses D + 9 + 99 and will be found by PEEK (D+T), where T has the value of 108. In that case, the subroutine at lines 108 to 124 will RETURN to line 114 or 124 with TT=9 and T=108 and assign value 8 to the PRINT AT line number variable X or Y. The difference between eight and three is explained by the tower tops being on screen line five.

The 'other' tower need not be PEEKed; the line number of its upper ring, Z, can be calculated because the total of lines from the tower tops to the upper rings on all three towers

is always twice the number of rings in play. Therefore, $X = 5 + Y - 5 + 2 - 5 = 2 \times N$ (check out lines 34 and 128).

HIGH TIMES

It will be seen that X, Y and Z are inverse measures of the heights of the piles of rings on the 'go from', 'go to' and other towers respectively.

$ABS(GF - GT) = 18$ means that the 'go from' and 'go to' towers are 18 column apart, so that the other tower is between them. If the height of its pile is greater than that of the GOTO tower, ie Z is less than Y, then the last (if any) of the moving ring must be

determined by Z and not Y — hence lines 134 to 136.

Programs with subroutines may suffer from lack of memory if they are frequently interrupted before all the GOSUBs have returned, unless return addresses accumulate in the GOSUB stack, causing it to go down in memory which may be required by the program itself. The effect of this is perpetuated when the program is SAVED.

In this program, the stack can be cleared in the following way. First add the following line:

50 RUN 91

Then, add the following two lines:

61 PRINT AT 14,0;PEEK
16386+256*PEEK
16387

81 PRINT AT 14,0;PEEK
16386+256*PEEK
16387

Finally, change line 82 to read:

82 RETURN

Repeat the command, RETURN, until the error report shows T82. Lines 61 and 81 are optional — they show the changes of the addresses of the GOSUB stack (it is 30764 when cleared). Don't forget to remove line 50 and restore line 82 before trying the game again though!

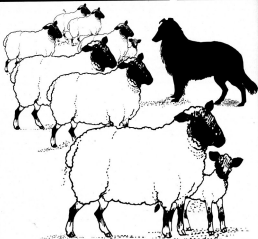
The listing for the program, Towers of Hanoi.

```

10 PRINT "TOWERS OF HANOI"
20 LET N=10
30 LET X=0
40 LET Y=0
50 LET Z=0
60 LET GF=0
70 LET GT=0
80 LET HT=0
90 LET H=0
100 LET M=0
110 LET S=0
120 LET D=0
130 LET E=0
140 LET F=0
150 LET G=0
160 LET H=0
170 LET I=0
180 LET J=0
190 LET K=0
200 LET L=0
210 LET M=0
220 LET N=0
230 LET O=0
240 LET P=0
250 LET Q=0
260 LET R=0
270 LET S=0
280 LET T=0
290 LET U=0
300 LET V=0
310 LET W=0
320 LET X=0
330 LET Y=0
340 LET Z=0
350 LET A=0
360 LET B=0
370 LET C=0
380 LET D=0
390 LET E=0
400 LET F=0
410 LET G=0
420 LET H=0
430 LET I=0
440 LET J=0
450 LET K=0
460 LET L=0
470 LET M=0
480 LET N=0
490 LET O=0
500 LET P=0
510 LET Q=0
520 LET R=0
530 LET S=0
540 LET T=0
550 LET U=0
560 LET V=0
570 LET W=0
580 LET X=0
590 LET Y=0
600 LET Z=0
610 LET A=0
620 LET B=0
630 LET C=0
640 LET D=0
650 LET E=0
660 LET F=0
670 LET G=0
680 LET H=0
690 LET I=0
700 LET J=0
710 LET K=0
720 LET L=0
730 LET M=0
740 LET N=0
750 LET O=0
760 LET P=0
770 LET Q=0
780 LET R=0
790 LET S=0
800 LET T=0
810 LET U=0
820 LET V=0
830 LET W=0
840 LET X=0
850 LET Y=0
860 LET Z=0
870 LET A=0
880 LET B=0
890 LET C=0
900 LET D=0
910 LET E=0
920 LET F=0
930 LET G=0
940 LET H=0
950 LET I=0
960 LET J=0
970 LET K=0
980 LET L=0
990 LET M=0
1000 LET N=0
1010 LET O=0
1020 LET P=0
1030 LET Q=0
1040 LET R=0
1050 LET S=0
1060 LET T=0
1070 LET U=0
1080 LET V=0
1090 LET W=0
1100 LET X=0
1110 LET Y=0
1120 LET Z=0
1130 LET A=0
1140 LET B=0
1150 LET C=0
1160 LET D=0
1170 LET E=0
1180 LET F=0
1190 LET G=0
1200 LET H=0
1210 LET I=0
1220 LET J=0
1230 LET K=0
1240 LET L=0
1250 LET M=0
1260 LET N=0
1270 LET O=0
1280 LET P=0
1290 LET Q=0
1300 LET R=0
1310 LET S=0
1320 LET T=0
1330 LET U=0
1340 LET V=0
1350 LET W=0
1360 LET X=0
1370 LET Y=0
1380 LET Z=0
1390 LET A=0
1400 LET B=0
1410 LET C=0
1420 LET D=0
1430 LET E=0
1440 LET F=0
1450 LET G=0
1460 LET H=0
1470 LET I=0
1480 LET J=0
1490 LET K=0
1500 LET L=0
1510 LET M=0
1520 LET N=0
1530 LET O=0
1540 LET P=0
1550 LET Q=0
1560 LET R=0
1570 LET S=0
1580 LET T=0
1590 LET U=0
1600 LET V=0
1610 LET W=0
1620 LET X=0
1630 LET Y=0
1640 LET Z=0
1650 LET A=0
1660 LET B=0
1670 LET C=0
1680 LET D=0
1690 LET E=0
1700 LET F=0
1710 LET G=0
1720 LET H=0
1730 LET I=0
1740 LET J=0
1750 LET K=0
1760 LET L=0
1770 LET M=0
1780 LET N=0
1790 LET O=0
1800 LET P=0
1810 LET Q=0
1820 LET R=0
1830 LET S=0
1840 LET T=0
1850 LET U=0
1860 LET V=0
1870 LET W=0
1880 LET X=0
1890 LET Y=0
1900 LET Z=0
1910 LET A=0
1920 LET B=0
1930 LET C=0
1940 LET D=0
1950 LET E=0
1960 LET F=0
1970 LET G=0
1980 LET H=0
1990 LET I=0
2000 LET J=0
2010 LET K=0
2020 LET L=0
2030 LET M=0
2040 LET N=0
2050 LET O=0
2060 LET P=0
2070 LET Q=0
2080 LET R=0
2090 LET S=0
2100 LET T=0
2110 LET U=0
2120 LET V=0
2130 LET W=0
2140 LET X=0
2150 LET Y=0
2160 LET Z=0
2170 LET A=0
2180 LET B=0
2190 LET C=0
2200 LET D=0
2210 LET E=0
2220 LET F=0
2230 LET G=0
2240 LET H=0
2250 LET I=0
2260 LET J=0
2270 LET K=0
2280 LET L=0
2290 LET M=0
2300 LET N=0
2310 LET O=0
2320 LET P=0
2330 LET Q=0
2340 LET R=0
2350 LET S=0
2360 LET T=0
2370 LET U=0
2380 LET V=0
2390 LET W=0
2400 LET X=0
2410 LET Y=0
2420 LET Z=0
2430 LET A=0
2440 LET B=0
2450 LET C=0
2460 LET D=0
2470 LET E=0
2480 LET F=0
2490 LET G=0
2500 LET H=0
2510 LET I=0
2520 LET J=0
2530 LET K=0
2540 LET L=0
2550 LET M=0
2560 LET N=0
2570 LET O=0
2580 LET P=0
2590 LET Q=0
2600 LET R=0
2610 LET S=0
2620 LET T=0
2630 LET U=0
2640 LET V=0
2650 LET W=0
2660 LET X=0
2670 LET Y=0
2680 LET Z=0
2690 LET A=0
2700 LET B=0
2710 LET C=0
2720 LET D=0
2730 LET E=0
2740 LET F=0
2750 LET G=0
2760 LET H=0
2770 LET I=0
2780 LET J=0
2790 LET K=0
2800 LET L=0
2810 LET M=0
2820 LET N=0
2830 LET O=0
2840 LET P=0
2850 LET Q=0
2860 LET R=0
2870 LET S=0
2880 LET T=0
2890 LET U=0
2900 LET V=0
2910 LET W=0
2920 LET X=0
2930 LET Y=0
2940 LET Z=0
2950 LET A=0
2960 LET B=0
2970 LET C=0
2980 LET D=0
2990 LET E=0
3000 LET F=0
3010 LET G=0
3020 LET H=0
3030 LET I=0
3040 LET J=0
3050 LET K=0
3060 LET L=0
3070 LET M=0
3080 LET N=0
3090 LET O=0
3100 LET P=0
3110 LET Q=0
3120 LET R=0
3130 LET S=0
3140 LET T=0
3150 LET U=0
3160 LET V=0
3170 LET W=0
3180 LET X=0
3190 LET Y=0
3200 LET Z=0
3210 LET A=0
3220 LET B=0
3230 LET C=0
3240 LET D=0
3250 LET E=0
3260 LET F=0
3270 LET G=0
3280 LET H=0
3290 LET I=0
3300 LET J=0
3310 LET K=0
3320 LET L=0
3330 LET M=0
3340 LET N=0
3350 LET O=0
3360 LET P=0
3370 LET Q=0
3380 LET R=0
3390 LET S=0
3400 LET T=0
3410 LET U=0
3420 LET V=0
3430 LET W=0
3440 LET X=0
3450 LET Y=0
3460 LET Z=0
3470 LET A=0
3480 LET B=0
3490 LET C=0
3500 LET D=0
3510 LET E=0
3520 LET F=0
3530 LET G=0
3540 LET H=0
3550 LET I=0
3560 LET J=0
3570 LET K=0
3580 LET L=0
3590 LET M=0
3600 LET N=0
3610 LET O=0
3620 LET P=0
3630 LET Q=0
3640 LET R=0
3650 LET S=0
3660 LET T=0
3670 LET U=0
3680 LET V=0
3690 LET W=0
3700 LET X=0
3710 LET Y=0
3720 LET Z=0
3730 LET A=0
3740 LET B=0
3750 LET C=0
3760 LET D=0
3770 LET E=0
3780 LET F=0
3790 LET G=0
3800 LET H=0
3810 LET I=0
3820 LET J=0
3830 LET K=0
3840 LET L=0
3850 LET M=0
3860 LET N=0
3870 LET O=0
3880 LET P=0
3890 LET Q=0
3900 LET R=0
3910 LET S=0
3920 LET T=0
3930 LET U=0
3940 LET V=0
3950 LET W=0
3960 LET X=0
3970 LET Y=0
3980 LET Z=0
3990 LET A=0
4000 LET B=0
4010 LET C=0
4020 LET D=0
4030 LET E=0
4040 LET F=0
4050 LET G=0
4060 LET H=0
4070 LET I=0
4080 LET J=0
4090 LET K=0
4100 LET L=0
4110 LET M=0
4120 LET N=0
4130 LET O=0
4140 LET P=0
4150 LET Q=0
4160 LET R=0
4170 LET S=0
4180 LET T=0
4190 LET U=0
4200 LET V=0
4210 LET W=0
4220 LET X=0
4230 LET Y=0
4240 LET Z=0
4250 LET A=0
4260 LET B=0
4270 LET C=0
4280 LET D=0
4290 LET E=0
4300 LET F=0
4310 LET G=0
4320 LET H=0
4330 LET I=0
4340 LET J=0
4350 LET K=0
4360 LET L=0
4370 LET M=0
4380 LET N=0
4390 LET O=0
4400 LET P=0
4410 LET Q=0
4420 LET R=0
4430 LET S=0
4440 LET T=0
4450 LET U=0
4460 LET V=0
4470 LET W=0
4480 LET X=0
4490 LET Y=0
4500 LET Z=0
4510 LET A=0
4520 LET B=0
4530 LET C=0
4540 LET D=0
4550 LET E=0
4560 LET F=0
4570 LET G=0
4580 LET H=0
4590 LET I=0
4600 LET J=0
4610 LET K=0
4620 LET L=0
4630 LET M=0
4640 LET N=0
4650 LET O=0
4660 LET P=0
4670 LET Q=0
4680 LET R=0
4690 LET S=0
4700 LET T=0
4710 LET U=0
4720 LET V=0
4730 LET W=0
4740 LET X=0
4750 LET Y=0
4760 LET Z=0
4770 LET A=0
4780 LET B=0
4790 LET C=0
4800 LET D=0
4810 LET E=0
4820 LET F=0
4830 LET G=0
4840 LET H=0
4850 LET I=0
4860 LET J=0
4870 LET K=0
4880 LET L=0
4890 LET M=0
4900 LET N=0
4910 LET O=0
4920 LET P=0
4930 LET Q=0
4940 LET R=0
4950 LET S=0
4960 LET T=0
4970 LET U=0
4980 LET V=0
4990 LET W=0
5000 LET X=0
5010 LET Y=0
5020 LET Z=0
5030 LET A=0
5040 LET B=0
5050 LET C=0
5060 LET D=0
5070 LET E=0
5080 LET F=0
5090 LET G=0
5100 LET H=0
5110 LET I=0
5120 LET J=0
5130 LET K=0
5140 LET L=0
5150 LET M=0
5160 LET N=0
5170 LET O=0
5180 LET P=0
5190 LET Q=0
5200 LET R=0
5210 LET S=0
5220 LET T=0
5230 LET U=0
5240 LET V=0
5250 LET W=0
5260 LET X=0
5270 LET Y=0
5280 LET Z=0
5290 LET A=0
5300 LET B=0
5310 LET C=0
5320 LET D=0
5330 LET E=0
5340 LET F=0
5350 LET G=0
5360 LET H=0
5370 LET I=0
5380 LET J=0
5390 LET K=0
5400 LET L=0
5410 LET M=0
5420 LET N=0
5430 LET O=0
5440 LET P=0
5450 LET Q=0
5460 LET R=0
5470 LET S=0
5480 LET T=0
5490 LET U=0
5500 LET V=0
5510 LET W=0
5520 LET X=0
5530 LET Y=0
5540 LET Z=0
5550 LET A=0
5560 LET B=0
5570 LET C=0
5580 LET D=0
5590 LET E=0
5600 LET F=0
5610 LET G=0
5620 LET H=0
5630 LET I=0
5640 LET J=0
5650 LET K=0
5660 LET L=0
5670 LET M=0
5680 LET N=0
5690 LET O=0
5700 LET P=0
5710 LET Q=0
5720 LET R=0
5730 LET S=0
5740 LET T=0
5750 LET U=0
5760 LET V=0
5770 LET W=0
5780 LET X=0
5790 LET Y=0
5800 LET Z=0
5810 LET A=0
5820 LET B=0
5830 LET C=0
5840 LET D=0
5850 LET E=0
5860 LET F=0
5870 LET G=0
5880 LET H=0
5890 LET I=0
5900 LET J=0
5910 LET K=0
5920 LET L=0
5930 LET M=0
5940 LET N=0
5950 LET O=0
5960 LET P=0
5970 LET Q=0
5980 LET R=0
5990 LET S=0
6000 LET T=0
6010 LET U=0
6020 LET V=0
6030 LET W=0
6040 LET X=0
6050 LET Y=0
6060 LET Z=0
6070 LET A=0
6080 LET B=0
6090 LET C=0
6100 LET D=0
6110 LET E=0
6120 LET F=0
6130 LET G=0
6140 LET H=0
6150 LET I=0
6160 LET J=0
6170 LET K=0
6180 LET L=0
6190 LET M=0
6200 LET N=0
6210 LET O=0
6220 LET P=0
6230 LET Q=0
6240 LET R=0
6250 LET S=0
6260 LET T=0
6270 LET U=0
6280 LET V=0
6290 LET W=0
6300 LET X=0
6310 LET Y=0
6320 LET Z=0
6330 LET A=0
6340 LET B=0
6350 LET C=0
6360 LET D=0
6370 LET E=0
6380 LET F=0
6390 LET G=0
6400 LET H=0
6410 LET I=0
6420 LET J=0
6430 LET K=0
6440 LET L=0
6450 LET M=0
6460 LET N=0
6470 LET O=0
6480 LET P=0
6490 LET Q=0
6500 LET R=0
6510 LET S=0
6520 LET T=0
6530 LET U=0
6540 LET V=0
6550 LET W=0
6560 LET X=0
6570 LET Y=0
6580 LET Z=0
6590 LET A=0
6600 LET B=0
6610 LET C=0
6620 LET D=0
6630 LET E=0
6640 LET F=0
6650 LET G=0
6660 LET H=0
6670 LET I=0
6680 LET J=0
6690 LET K=0
6700 LET L=0
6710 LET M=0
6720 LET N=0
6730 LET O=0
6740 LET P=0
6750 LET Q=0
6760 LET R=0
6770 LET S=0
6780 LET T=0
6790 LET U=0
6800 LET V=0
6810 LET W=0
6820 LET X=0
6830 LET Y=0
6840 LET Z=0
6850 LET A=0
6860 LET B=0
6870 LET C=0
6880 LET D=0
6890 LET E=0
6900 LET F=0
6910 LET G=0
6920 LET H=0
6930 LET I=0
6940 LET J=0
6950 LET K=0
6960 LET L=0
6970 LET M=0
6980 LET N=0
6990 LET O=0
7000 LET P=0
7010 LET Q=0
7020 LET R=0
7030 LET S=0
7040 LET T=0
7050 LET U=0
7060 LET V=0
7070 LET W=0
7080 LET X=0
7090 LET Y=0
7100 LET Z=0
7110 LET A=0
7120 LET B=0
7130 LET C=0
7140 LET D=0
7150 LET E=0
7160 LET F=0
7170 LET G=0
7180 LET H=0
7190 LET I=0
7200 LET J=0
7210 LET K=0
7220 LET L=0
7230 LET M=0
7240 LET N=0
7250 LET O=0
7260 LET P=0
7270 LET Q=0
7280 LET R=0
7290 LET S=0
7300 LET T=0
7310 LET U=0
7320 LET V=0
7330 LET W=0
7340 LET X=0
7350 LET Y=0
7360 LET Z=0
7370 LET A=0
7380 LET B=0
7390 LET C=0
7400 LET D=0
7410 LET E=0
7420 LET F=0
7430 LET G=0
7440 LET H=0
7450 LET I=0
7460 LET J=0
7470 LET K=0
7480 LET L=0
7490 LET M=0
7500 LET N=0
7510 LET O=0
7520 LET P=0
7530 LET Q=0
7540 LET R=0
7550 LET S=0
7560 LET T=0
7570 LET U=0
7580 LET V=0
7590 LET W=0
7600 LET X=0
7610 LET Y=0
7620 LET Z=0
7630 LET A=0
7640 LET B=0
7650 LET C=0
7660 LET D=0
7670 LET E=0
7680 LET F=0
7690 LET G=0
7700 LET H=0
7710 LET I=0
7720 LET J=0
7730 LET K=0
7740 LET L=0
7750 LET M=0
7760 LET N=0
7770 LET O=0
7780 LET P=0
7790 LET Q=0
7800 LET R=0
7810 LET S=0
7820 LET T=0
7830 LET U=0
7840 LET V=0
7850 LET W=0
7860 LET X=0
7870 LET Y=0
7880 LET Z=0
7890 LET A=0
7900 LET B=0
7910 LET C=0
7920 LET D=0
7930 LET E=0
7940 LET F=0
7950 LET G=0
7960 LET H=0
7970 LET I=0
7980 LET J=0
7990 LET K=0
8000 LET L=0
8010 LET M=0
8020 LET N=0
8030 LET O=0
8040 LET P=0
8050 LET Q=0
8060 LET R=0
8070 LET S=0
8080 LET T=0
8090 LET U=0
8100 LET V=0
8110 LET W=0
8120 LET X=0
8130 LET Y=0
8140 LET Z=0
8150 LET A=0
8160 LET B=0
8170 LET C=0
8180 LET D=0
8190 LET E=0
8200 LET F=0
8210 LET G=0
8220 LET H=0
8230 LET I=0
8240 LET J=0
8250 LET K=0
8260 LET L=0
8270 LET M=0
8280 LET N=0
8290 LET O=0
8300 LET P=0
8310 LET Q=0
8320 LET R=0
8330 LET S=0
8340 LET T=0
8350 LET U=0
8360 LET V=0
8370 LET W=0
8380 LET X=0
8390 LET Y=0
8400 LET Z=0
8410 LET A=0
8420 LET B=0
8430 LET C=0
8440 LET D=0
8450 LET E=0
8460 LET F=0
8470 LET G=0
8480 LET H=0
8490 LET I=0
8500 LET J=0
8510 LET K=0
8520 LET L=0
8530 LET M=0
8540 LET N=0
8550 LET O=0
8560 LET P=0
8570 LET Q=0
8580 LET R=0
8590 LET S=0
8600 LET T=0
8610 LET U=0
8620 LET V=0
8630 LET W=0
8640 LET X=0
8650 LET Y=0
8660 LET Z=0
8670 LET A=0
8680 LET B=0
8690 LET C=0
8700 LET D=0
8710 LET E=0
8720 LET F=0
8730 LET G=0
8740 LET H=0
8750 LET I=0
8760 LET J=0
8770 LET K=0
8780 LET L=0
8790 LET M=0
8800 LET N=0
8810 LET O=0
8820 LET P=0
8830 LET Q=0
8840 LET R=0
8850 LET S=0
8860 LET T=0
8870 LET U=0
8880 LET V=0
8890 LET W=0
8900 LET X=0
8910 LET Y=0
8920 LET Z=0
8930 LET A=0
8940 LET B=0
8950 LET C=0
8960 LET D=0
8970 LET E=0
8980 LET F=0
8990 LET G=0
9000 LET H=0
9010 LET I=0
9020 LET J=0
9030 LET K=0
9040 LET L=0
9050 LET M=0
9060 LET N=0
9070 LET O=0
9080 LET P=0
9090 LET Q=0
9100 LET R=0
9110 LET S=0
9120 LET T=0
9130 LET U=0
9140 LET V=0
9150 LET W=0
9160 LET X=0
9170 LET Y=0
9180 LET Z=0
9190 LET A=0
9200 LET B=0
9210 LET C=0
9220 LET D=0
9230 LET E=0
9240 LET F=0
9250 LET G=0
9260 LET H=0
9270 LET I=0
9280 LET J=0
9290 LET K=0
9300 LET L=0
9310 LET M=0
9320 LET N=0
9330 LET O=0
9340 LET P=0
9350 LET Q=0
9360 LET R=0
9370 LET S=0
9380 LET T=0
9390 LET U=0
9400 LET V=0
9410 LET W=0
9420 LET X=0
9430 LET Y=0
9440 LET Z=0
9450 LET A=0
9460 LET B=0
9470 LET C=0
9480 LET D=0
9490 LET E=0
9500 LET F=0
9510 LET G=0
9520 LET H=0
9530 LET I=0
9540 LET J=0
9550 LET K=0
9560 LET L=0
9570 LET M=0
9580 LET N=0
9590 LET O=0
9600 LET P=0
9610 LET Q=0
9620 LET R=0
9630 LET S=0
9640 LET T=0
9650 LET U=0
9660 LET V=0
9670 LET W=0
9680 LET X=0
9690 LET Y=0
9700 LET Z=0
9710 LET A=0
9720 LET B=0
9730 LET C=0
9740 LET D=0
9750 LET E=0
9760 LET F=0
9770 LET G=0
9780 LET H=0
9790 LET I=0
9800 LET J=0
9810 LET K=0
9820 LET L=0
9830 LET M=0
9840 LET N=0
9850 LET O=0
9860 LET P=0
9870 LET Q=0
9880 LET R=0
9890 LET S=0
9900 LET T=0
9910 LET U=0
9920 LET V=0
9930 LET W=0
9940 LET X=0
9950 LET Y=0
9960 LET Z=0
9970 LET A=0
9980 LET B=0
9990 LET C=0
10000 LET D=0
10010 LET E=0
10020 LET F=0
10030 LET G=0
10040 LET H=0
10050 LET I=0
10060 LET J=0
10070 LET K=0
10080 LET L=0
10090 LET M=0
10100 LET N=0
10110 LET O=0
10120 LET P=0
10130 LET Q=0
10140 LET R=0
10150 LET S=0
10160 LET T=0
10170 LET U=0
10180 LET V=0
10190 LET W=0
10200 LET X=0
10210 LET Y=0
10220 LET Z=0
10230 LET A=0
10240 LET B=0
10250 LET C=0
10260 LET D=0
10270 LET E=0
10280 LET F=0
10290 LET G=0
10300 LET H=0
10310 LET I=0
10320 LET J=0
10330 LET K=0
10340 LET L=0
10350 LET M=0
10360 LET N=0
10370 LET O=0
10380 LET P=0
10390 LET Q=0
10400 LET R=0
10410 LET S=0
10420 LET T=0
10430 LET U=0
10440 LET V=0
10450 LET W=0
10460 LET X=0
10470 LET Y=0
10480 LET Z=0
10490 LET A=0
10500 LET B=0
10510 LET C=0
10520 LET D=0
10530 LET E=0
10540 LET F=0
10550 LET G=0
10560 LET H=0
10570 LET I=0
10580 LET J=0
10590 LET K=0
10600 LET L=0
10610 LET M=0
10620 LET N=0
10630 LET O=0
10640 LET P=0
10650 LET Q=0
10660 LET R=0
10670 LET S=0
10680 LET T=0
10690 LET U=0
10700 LET V=0
10710 LET W=0
10720 LET X=0
10730 LET Y=0
10740 LET Z=0
10750 LET A=0
10760 LET B=0
10770 LET C=0
10780 LET D=0
10790 LET E=0
10800 LET F=0
10810 LET G=0
10820 LET H=0
10830 LET I=0
10840 LET J=0
10850 LET K=0
10860 LET L=0
10870 LET M=0
10880 LET N=0
10890 LET O=0
10900 LET P=0
10910 LET Q=0
10920 LET R=0
10930 LET S=0
10940 LET T=0
10950 LET U=0
10960 LET V=0
10970 LET W=0
10980 LET X=0
10990 LET Y=0
11000 LET Z=0
11010 LET A=0
11020 LET B=0
11030 LET C=0
11040 LET D=0
11050 LET E=0
11060 LET F=0
11070 LET G=0
11080 LET H=0
11090 LET I=0
11100 LET J=0
11110 LET K=0
11120 LET L=0
11130 LET M=0
11140 LET N=0
11150 LET O=0
11160 LET P=0
11170 LET Q=0
11180 LET R=0
11190 LET S=0
11200 LET T=0
11210 LET U=0
11220 LET V=0
11230 LET W=0
11240 LET X=0
11250 LET Y=0
11260 LET Z=0
11270 LET A=0
11280 LET B=0
11290 LET C=0
11300 LET D=0
11310 LET E=0
11320 LET F=0
11330 LET G=0
11340 LET H=0
11350 LET I=0
11360 LET J=0
11370 LET K=0
11380 LET L=0
11390 LET M=0
11400 LET N=0
11410 LET O=0
11420 LET P=0
11430 LET Q=0
11440 LET R=0
11450 LET S=0
11460 LET T=0
11470 LET U=0
11480 LET V=0
11490 LET W=0
11500 LET X=0
11510 LET Y=0
11520 LET Z=0
11530 LET A=0
11540 LET B=0
11550 LET C=0
11560 LET D=0
11570 LET E=0
11580 LET F=0
11590 LET G=0
11600 LET H=0
11610 LET I=0
11620 LET J=0
11630 LET K=0
11640 LET L=0
11650 LET M=0
11660 LET N=0
11670 LET O=0
11680 LET P=0
11690 LET Q=0
11700 LET R=0
11710 LET S=0
11720 LET T=0
11730 LET U=0
11740 LET V=0
11750 LET W=0
11760 LET X=0
11770 LET Y=0
11780 LET Z=0
11790 LET A=0
11800 LET B=0
11810 LET C=0
11820 LET D=0
11830 LET E=0
11840 LET F=0
11850 LET G=0
11860 LET H=0
11870 LET I=0
11880 LET J=0
11890 LET K=0
11900 LET L=0
11910 LET M=0
11920 LET N=0
11930 LET O=0
11940 LET P=0
11950 LET Q=0
11960 LET R=0
11970 LET S=0
11980 LET T=0
11990 LET U=0
12000 LET V=0
12010 LET W=0
12020 LET X=0
12030 LET Y=0
12040 LET Z=0
12050 LET A=0
12060 LET B=0
12070 LET C=0
12080 LET D=0
12090 LET E=0
12100 LET F=0
12110 LET G=0
12120 LET H=0
12130 LET I=0
12140 LET J=0
12150 LET K=0
12160 LET L=0
12170 LET M=0
12180 LET N=0
12190 LET O=0
12200 LET P=0
12210 LET Q=0
12220 LET R=0
12230 LET S=0
12240 LET T=0
12250 LET U=0
12260 LET V=0
12270 LET W=0
12280 LET X=0
12290 LET Y=0
12300 LET Z=0
12310 LET A=0
12320 LET B=0
12330 LET C=0
12340 LET D=0
12350 LET E=0
12360 LET F=0
12370 LET G=0
12380 LET H=0
12390 LET I=0
12400 LET J=0
12410 LET K=0
12420 LET L=0
12430 LET M
```

SHEEPDOG TRIALS

Round up your sheep
but don't get penned in!



The object of this game is to round up a number of sheep in the shortest possible time. You are given the choice of how many sheep (between one and six) you would like to round up; one sheep is fairly easy, but if you choose to round up six sheep you will find it very time consuming as they keep wandering out of the pen as others are being rounded up.

You move the sheep by placing the dog near to the sheep using the cursor keys, so that it will move directly away from the dog. However, if you move too close to the sheep, you may find that the sheep will panic and are liable to move in an unpredictable direction.

The dog is allowed three moves between sheep moves; this can be increased if required by changing line 710.

The game ends when the dog has rounded up the sheep and is standing guard at the gate of the pen and the dog's three moves have been completed. At any stage of the game, neither the dog or sheep are allowed to jump the fence or wall of the pen.

IT'S A DOG'S LIFE

The dog and sheep are **POKED**

onto the screen and the previous positions of the characters are blanked by POKEing with zero. The POKE numbers are calculated as offsets using the number held in the system variable D-FILE as a base point.

The checks on the relationship between sheep and

dog are made in terms of x and y co-ordinates, so relatives to convert to these are included; they are located at the beginning of the program to speed up the operation. The routine to generate random sheep moves is also placed at the beginning of the program for the same reason. The

positions of the sheep in terms of the POKE offsets are held in the array, S(43).

The timer, which is initialized in lines 700 to 709 and updated by the subroutine at line 2000, makes use of the FRAMES system variable. Lines 5000 and 5005 are the usual Z801 self-RUN routine.

The listing for Sheepdog Trial.

```

1 REM *****SHEEPDOG TRIAL***
2 REM *****
30 PRINT AT 0.10;"SHEEPDOG TRI
40
50 PRINT AT 0.0;"YOU MAY CHOOSE
60 HOW MANY SHEEP TO ROUND UP.
70 NO MORE THAN 5. IF YOU GET TOO
80 CLOSE TO 5 SHEEP, I WILL RUN
90 IT AND IF YOU SEE 100, FOR RUN
100 SHEEP WILL UNDER AT 1, RANDOM.
110
120 PRINT "HOW DOG WITH THE CU
130 RSED KEY?A- TO USE A TURN ULTRU
140 U- HOLDING. THE DOG PRESS ANY OF
150 HER KEY.
160 PRINT AT 13.0;"PRESS ANY KE
170 Y TO PLAY.
180 IF INKEY$="" THEN GOTO 40
190 CLS
200 GOTO 300
210 REM *****CPU. TO CO-ORDS***
220 LET YS=INT (2/23)+1
230 LET XS=12-(YS-1)*33
240 RETURN
250 REM *****CO-ORDS***
260 LET YS=INT (2/23)+1
270 LET XS=12-(YS-1)*33
280 RETURN
290 REM *****RANDOM MOVES***
300 LET XS=X$+1-INT (RND*3)
310 LET YS=Y$+1-INT (RND*3)
320 GOTO 300
330 REM *****
340 LET US=X$-33+YS-1
350 RETURN
360 REM *****OF SHEEP***
370 PRINT "HOW MANY SHEEP DO YO
380 U WANT TO ROUND UP?
390 INPUT M
400 IF M<0 AND M>7 THEN GOTO
410
420 PRINT "SORRY. MORE THAN 8
430 SHEEP.
440 GOTO 310
450 PRINT "OK. IS OKAY."
460 FOR M=1 TO 100
470 NEXT M
480 GOTO
490 *****INITIALIZE***
500 REM *****
510 GOTO 310
520 REM *****
530 PRINT AT 0.13;
540 PRINT AT 10.13;
550 PRINT AT 11.13;
560 PRINT AT 12.13;
570 REM *****
580 FOR M=0 TO 31
590 PRINT AT 0.M;" "AT 31.M".
600
610 NEXT M
620 FOR M=0 TO 31
630 PRINT AT M.0;" "AT M.31".
640
650 NEXT M
660 REM *****GENERATE SHEEP***
670 LET U=PEEK 16380+255*PEEK 1
680
690 FOR M=1 TO M$
700 LET US=INT (RND*724)
710 IF PEEK (U+US)*10 THEN GOTO
720
730 POKE U+US,20
740 LET M=INT US
750 NEXT M
760 REM *****CLOSE DOGS***
770 LET U=INT (RND*724)
780 IF PEEK (U+U)*10 THEN GOTO
790
800 POKE U+U,0
810 NEXT M
820 REM *****
830 GOTO 1
840
850 REM *****
860 LET U=INT (2/23)+1
870 LET XS=12-(U-1)*33
880 RETURN
890
900 REM *****
910 GOTO 1

```

USER CHARACTER SET



Strailer in nature to the user definable graphics program by Chris Calender which appeared in the August/September issue of *ZX Computing*, this program is rather more flexible in that it will store a character set of up to 39 characters and enable them to be LPRINTed simply by entering a string (X\$).

Once you have entered the program, you will be greeted with the prompt:

NOW ENTER THE CODES FOR THE SPECTRUM CHARACTER SET

Simply type in the codes given for the lower case characters (like on the Spectrum), and these will be placed in the array, U.

You will now be given the prompt:

ENTER TEXT TO BE LPRINTED USING NEW CHARACTER SET

At this stage, you simply enter what you would like to be printed in upper and lower case.

Any letters you enter as normal video will be printed in

lower case. Any letters entered as inverse video will be printed as upper case. Numbers entered as inverse video will actually be printed as normal video numbers, whereas if any numbers are entered as normal video, they will be printed as the extra characters which have been defined for them.

EXTRA, EXTRA

If you have entered your own characters, having answered 'Y' to the prompt:

DO YOU WANT TO ENTER YOUR OWN USER-DEFINED CHARACTERS? (Y OR N)

then they can be obtained by typing in the graphics corresponding to the Sinclair codes 1 to 10 (for details check out Appendix A of the Sinclair ZX81 manual). For example, CHR\$ 1 corresponds to the first user-definable graphic.

Character definition should only be attempted if you understand the way that a character is made up from binary numbers (eight for each character). These numbers must then be translated into decimal

Print upper and lower case characters with your ZX81.

to be typed in to the program. The extra graphics characters, along with the number that has to be entered to produce them is shown in Fig. 1.

LINE BY LINE

Before typing in or re-LOADing this program from tape, it is important that you enter the following two lines (each followed by Newline) as direct commands:

**POKE 16389,124
NEW**

(in order to reserve space above RAMtop for the LPRINT routine which is copied out from the ROM in lines 10 to 50).

Here follows a brief breakdown of the structure of the program.

Lines 100 to 130 copy from the ROM into the array S, all the codes for Sinclair's characters, from 0 to Z (normal video). Lines 200 to 230 copy from the ROM into the array U, all the extra Sinclair symbols, such as " , ' etc. These will then be obtainable directly when the program is in use. Lines 300 to 330 allow for the user to enter the codes for the lower case characters into the array U. (These codes are shown in Fig. 1). Every other line of this printout has the codes to be entered, separated by 'Y' to represent Newline, and above each set of codes is the number that should appear at the top of the screen when these codes are to be entered (and represent the codes' positions in the array U). Lines 500 to 700 allow you to enter wherever you wish to be printed using the new character

set, and then looks up the codes for these characters in the appropriate arrays, mapping them into the array A, for LPRINTING using the subroutine from Sinclair's printer manual (lines 9990 to 9999). When all

the arrays have been entered, lines 100 to 900 can be deleted. Line 1 should then be changed to read I SAVE "Character". Connected up your cassette, start it recording, then enter GOTO 1 as a direct command. Do not

use RUN or the arrays will be wiped. Make sure that all DIM statements have been entered. The program will begin straight away, which avoids the danger of the user typing RUN instead of GOTO to begin.

```

80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

300 POKE 31744+I,PEEK (2161+J)
301 NEXT I
400 POKE 31000,0
500 POKE 31007,255
1000 DIM S(255,0)
1100 LET X=7500
1200 FOR P=1 TO 25
1300 FOR N=1 TO 8
1400 LET S(N,P)=PEEK (X+P)
1500 NEXT P
1600 LET X=X+8
1700 NEXT N
2000 DIM U(25,0)
2100 LET X=7700
2200 FOR N=1 TO 27
2300 FOR P=1 TO 8
2400 LET U(N,P)=PEEK (X+P)
2500 NEXT P
2600 LET X=X+8
2700 NEXT N
2800 GOTO 888
2900 PRINT AT 3,0,"NOW ENTER THE
S CODES FOR THE SPECTRUM CHAR-
ACTER SET"
3100 FOR N=0 TO 63
3200 PRINT AT 3,0,N
3300 FOR P=1 TO 8
3400 PRINT AT 4,0,P
3500 INPUT U(N,P)
3600 NEXT P
3700 NEXT N
3800 CLS
3900 PRINT "ADD YOU WANT TO ENTER
S YOUR OWN USER-DEFINED CHARACTER
SET? Y OR N?"
4000 INPUT A$
4100 IF A$="N" THEN GOTO 500
4200 FOR N=1 TO 10
4300 PRINT AT 3,0,N
4400 FOR P=1 TO 8
4500 PRINT AT 4,0,P
4600 INPUT U(N,P)
4700 NEXT P
4800 NEXT N
4900 CLS
5000 SLOW
5100 PRINT "ENTERED TEXT TO BE LFN
ENTERED USING NEW CHARACTER SET."
5200 INPUT X$
5300 PRINT X$
5400 DIM R(10,0)
5500 FOR A=1 TO LEN X$
5600 IF A$[A]="" THEN GOTO 700
5700 IF CODE X$[A]<127 THEN GOTO
5800 FOR Y=1 TO 8
5900 LET R(A,Y)=CODE X$[A]+Y
6000 NEXT Y
6100 GOTO 700
6200 FOR Y=1 TO 8
6300 LET A=A,Y=CODE X$[A]-127
6400 NEXT Y
6500 NEXT A
6600 FOR J=1 TO 32
6700 FOR N=1 TO 8
6800 POKE 32000+N*8+(J-1)*81,U(N,J)
6900 NEXT N
7000 NEXT J
7100 FOR M=0 TO 31
7200 POKE 32444-M,M
7300 NEXT M
7400 LET HEIGHT=USR 31744
7500 GOTO 500

```

Fig. 1. The codes for the lower case characters - to be entered into the array, U.

```

1 SAVE "CHARACTER"
2 PRINT
10 FOR I=0 TO 115

```

THE SHARP END



Execution is a graphic variation on the anagrams game. The computer chooses a word randomly from its vocabulary, and you have to guess what the word is, by entering letters one at a time.

You're allowed 10 incorrect guesses before the blade on the guillotine does its job. There is no need to press NEWLINE when entering guesses.

Before you run the program the first time, enter DIM D\$(30,10) as a direct command, then start it with GOTO 1 (not RUN). Then, you'll be asked to enter 30 words, one at a time. For subsequent runs, start with GOTO 3, and your vocabulary will be safe. Enter "N" if you do not want another game at the end. **SAVE** the program by entering GOTO 300. Note that the inverse E in line 390 should be entered as an ordinary E.

ANAGRAMS

The Z88 randomly selects and scrambles a word. You must repeatedly enter what you think is the correct word, and the computer will underline those letters which are in the correct position.

You have a maximum number of attempts before the computer stops you and reveals the word. You use "N" if you don't want another game, or just press NEWLINE/ENTER if you do.

Entering the initial vocabulary, and the subsequent changing of it, is the same as in

Execution. Save the program by entering GOTO 340, and restart the program without losing your vocabulary, by entering GOTO 10. You'll need to set up an array, by entering DIM D\$(30,10) in the direct mode before you begin, then start with GOTO 1.

It takes some time to assemble a word, so don't be surprised if nothing happens for the first few seconds of the game.

CRASHER

Nothing to do with some computers' habits of blanking out! You are drifting in space, the object being to clear up as much debris as you can, particularly those with a high CODE value. The debris is valued according to the code of the character of debris. The letters score highest, then numbers and so on. Ignore inverse characters if they appear for they do not score because they've been affected by solar radiation and mutated to their inverse form, so are not worth anything. The keys 5 and 8 steer you left and right in the direction of the arrows on the keys. You score when the debris is directly ahead of you on screen so you "crash" into it. The position of your spaceship on screen is set by X (horizontal position) and Y (vertical position). They are initially set by lines 10 and 20 to a position at about the middle of the playing area of the screen. The variable S records your score

Execution and Anagrams are two great word games from Paul for the 16K ZX81. But if you have only 1K try **Crasher** from Dilwyn.

and is set to 0 at the start of the game by line 31. F is the variable that controls the duration of the game — it does not count in any particular unit but is a convenient way of controlling the length of the game. The time is clocked up in line 50, one unit of space-time of a time!! The statement LET F=F+... may confuse you a little since it is hard to imagine F being equal to F+1.

TRUTH

It actually means add 1 to the old value of F to make the new value. Line 60 is rather complicated. It determines where the character is PRINTED, which character is PRINTED and whether it's an inverse character. Let us look at the position first. The character is placed at the bottom of the playing area (the Y co-ordinate is set at 20 and X co-ordinate to a value from 0 to 19 by the random number expression. The expression after the semi-colon generates a character at random from the number generated in brackets after CHR\$. The number generated is a random number from 1 to 63. The following expression may look rather strange, but all it does is determine whether to add 0 or 128 to this number (to determine whether the character generated is inverse or not. It is a special use of the function AND. What it does is look at the following expression, add 0 to the value if it is not true, or the value before AND if it is true, so that 128 is added to the random number only if RND is less than .2. This makes it roughly a one in five chance of the character being an (inverse, non-scoring character. The statement after this crashes the old position of the Y before it is scrolled up the screen by line

70. This ensures the V is not PRINTed anywhere other than the middle of the screen. Line 80 checks the keyboard to see whether you're steering to the left, right, or keeping it stationary, and changes the value of X accordingly.

PRINTING AND RUSHING

The V is PRINTed in its new position in line 90. You can see how the effect given is that the

spaceship (V) remains stationary while space seems to rush past. The second part of line 90 moves the PRINT position to immediately ahead of the position of the spaceship.

Therefore we may find what is immediately in front of the spaceship by means of the line 100, which finds the CODE of the character stored at that memory location by SEEKing the address held in the system variables 16306 and 16309. Line 110 checks to see if this

character ahead of the spaceship by means of the line 66 if it has a CODE of less than 64 rather than less than 128 as you might expect from an inverse character detector). This would permit CHR\$ 118 NEWLINE markers that SCROLL might push up the screen. You may like to add this line which will stop the program if you hit a radiation-related piece of debris:

```
115 IF P > 127 THEN STOP
```

EXECUTION LISTING

```
100 FOR I=1 TO 30
101 INPUT S$(1);
102 NEXT I
103 PRINT " "
104 PRINT AT 2,10;"EXECUTION"
105 PRINT AT 2,0;
106 REMO
107 LET C=0
108 FOR I=1 TO 10
109 PRINT S;TAB 2;" "
110 NEXT I
111 PRINT AT 2,0;" "
112 PRINT AT 2,0;" "
113 PRINT AT 2,0;" "
114 LET US=1
115 FOR I=1 TO 10
116 IF C=0 THEN IF I=1 THEN LET US=
US+0.01;
117 NEXT I
118 LET US=1
119 PRINT AT 2,0;"THE WORD"
120 LET C=1
121 FOR I=1 TO 10
122 PRINT AT 10,20;C;AT 21,0;
123 GOTO 124
124 LET C=1
125 PRINT AT 2,0;" "
126 FOR I=1 TO 10
127 IF C=1 THEN GOTO 128
128 LET C=1
129 PRINT AT 2,0;" "
130 NEXT I
131 GOTO 132
132 LET C=1
133 PRINT AT 2,0;" "
134 NEXT I
135 GOTO 136
136 LET C=1
137 PRINT AT 2,0;" "
138 NEXT I
139 GOTO 140
140 LET C=1
141 PRINT AT 2,0;" "
142 NEXT I
143 GOTO 144
144 LET C=1
145 PRINT AT 2,0;" "
146 NEXT I
147 GOTO 148
148 LET C=1
149 PRINT AT 2,0;" "
150 NEXT I
151 GOTO 152
152 LET C=1
153 PRINT AT 2,0;" "
154 NEXT I
155 GOTO 156
156 LET C=1
157 PRINT AT 2,0;" "
158 NEXT I
159 GOTO 160
160 LET C=1
161 PRINT AT 2,0;" "
162 NEXT I
163 GOTO 164
164 LET C=1
165 PRINT AT 2,0;" "
166 NEXT I
167 GOTO 168
168 LET C=1
169 PRINT AT 2,0;" "
170 NEXT I
171 GOTO 172
172 LET C=1
173 PRINT AT 2,0;" "
174 NEXT I
175 GOTO 176
176 LET C=1
177 PRINT AT 2,0;" "
178 NEXT I
179 GOTO 180
180 LET C=1
181 PRINT AT 2,0;" "
182 NEXT I
183 GOTO 184
184 LET C=1
185 PRINT AT 2,0;" "
186 NEXT I
187 GOTO 188
188 LET C=1
189 PRINT AT 2,0;" "
190 NEXT I
191 GOTO 192
192 LET C=1
193 PRINT AT 2,0;" "
194 NEXT I
195 GOTO 196
196 LET C=1
197 PRINT AT 2,0;" "
198 NEXT I
199 GOTO 200
200 LET C=1
201 PRINT AT 2,0;" "
202 NEXT I
203 GOTO 204
204 LET C=1
205 PRINT AT 2,0;" "
206 NEXT I
207 GOTO 208
208 LET C=1
209 PRINT AT 2,0;" "
210 NEXT I
211 GOTO 212
212 LET C=1
213 PRINT AT 2,0;" "
214 NEXT I
215 GOTO 216
216 LET C=1
217 PRINT AT 2,0;" "
218 NEXT I
219 GOTO 220
220 LET C=1
221 PRINT AT 2,0;" "
222 NEXT I
223 GOTO 224
224 LET C=1
225 PRINT AT 2,0;" "
226 NEXT I
227 GOTO 228
228 LET C=1
229 PRINT AT 2,0;" "
230 NEXT I
231 GOTO 232
232 LET C=1
233 PRINT AT 2,0;" "
234 NEXT I
235 GOTO 236
236 LET C=1
237 PRINT AT 2,0;" "
238 NEXT I
239 GOTO 240
240 LET C=1
241 PRINT AT 2,0;" "
242 NEXT I
243 GOTO 244
244 LET C=1
245 PRINT AT 2,0;" "
246 NEXT I
247 GOTO 248
248 LET C=1
249 PRINT AT 2,0;" "
250 NEXT I
251 GOTO 252
252 LET C=1
253 PRINT AT 2,0;" "
254 NEXT I
255 GOTO 256
256 LET C=1
257 PRINT AT 2,0;" "
258 NEXT I
259 GOTO 260
260 LET C=1
261 PRINT AT 2,0;" "
262 NEXT I
263 GOTO 264
264 LET C=1
265 PRINT AT 2,0;" "
266 NEXT I
267 GOTO 268
268 LET C=1
269 PRINT AT 2,0;" "
270 NEXT I
271 GOTO 272
272 LET C=1
273 PRINT AT 2,0;" "
274 NEXT I
275 GOTO 276
276 LET C=1
277 PRINT AT 2,0;" "
278 NEXT I
279 GOTO 280
280 LET C=1
281 PRINT AT 2,0;" "
282 NEXT I
283 GOTO 284
284 LET C=1
285 PRINT AT 2,0;" "
286 NEXT I
287 GOTO 288
288 LET C=1
289 PRINT AT 2,0;" "
290 NEXT I
291 GOTO 292
292 LET C=1
293 PRINT AT 2,0;" "
294 NEXT I
295 GOTO 296
296 LET C=1
297 PRINT AT 2,0;" "
298 NEXT I
299 GOTO 300
300 LET C=1
301 PRINT AT 2,0;" "
302 NEXT I
303 GOTO 304
304 LET C=1
305 PRINT AT 2,0;" "
306 NEXT I
307 GOTO 308
308 LET C=1
309 PRINT AT 2,0;" "
310 NEXT I
311 GOTO 312
312 LET C=1
313 PRINT AT 2,0;" "
314 NEXT I
315 GOTO 316
316 LET C=1
317 PRINT AT 2,0;" "
318 NEXT I
319 GOTO 320
320 LET C=1
321 PRINT AT 2,0;" "
322 NEXT I
323 GOTO 324
324 LET C=1
325 PRINT AT 2,0;" "
326 NEXT I
327 GOTO 328
328 LET C=1
329 PRINT AT 2,0;" "
330 NEXT I
331 GOTO 332
332 LET C=1
333 PRINT AT 2,0;" "
334 NEXT I
335 GOTO 336
336 LET C=1
337 PRINT AT 2,0;" "
338 NEXT I
339 GOTO 340
340 LET C=1
341 PRINT AT 2,0;" "
342 NEXT I
343 GOTO 344
344 LET C=1
345 PRINT AT 2,0;" "
346 NEXT I
347 GOTO 348
348 LET C=1
349 PRINT AT 2,0;" "
350 NEXT I
351 GOTO 352
352 LET C=1
353 PRINT AT 2,0;" "
354 NEXT I
355 GOTO 356
356 LET C=1
357 PRINT AT 2,0;" "
358 NEXT I
359 GOTO 360
360 LET C=1
361 PRINT AT 2,0;" "
362 NEXT I
363 GOTO 364
364 LET C=1
365 PRINT AT 2,0;" "
366 NEXT I
367 GOTO 368
368 LET C=1
369 PRINT AT 2,0;" "
370 NEXT I
371 GOTO 372
372 LET C=1
373 PRINT AT 2,0;" "
374 NEXT I
375 GOTO 376
376 LET C=1
377 PRINT AT 2,0;" "
378 NEXT I
379 GOTO 380
380 LET C=1
381 PRINT AT 2,0;" "
382 NEXT I
383 GOTO 384
384 LET C=1
385 PRINT AT 2,0;" "
386 NEXT I
387 GOTO 388
388 LET C=1
389 PRINT AT 2,0;" "
390 NEXT I
391 GOTO 392
392 LET C=1
393 PRINT AT 2,0;" "
394 NEXT I
395 GOTO 396
396 LET C=1
397 PRINT AT 2,0;" "
398 NEXT I
399 GOTO 400
400 LET C=1
401 PRINT AT 2,0;" "
402 NEXT I
403 GOTO 404
404 LET C=1
405 PRINT AT 2,0;" "
406 NEXT I
407 GOTO 408
408 LET C=1
409 PRINT AT 2,0;" "
410 NEXT I
411 GOTO 412
412 LET C=1
413 PRINT AT 2,0;" "
414 NEXT I
415 GOTO 416
416 LET C=1
417 PRINT AT 2,0;" "
418 NEXT I
419 GOTO 420
420 LET C=1
421 PRINT AT 2,0;" "
422 NEXT I
423 GOTO 424
424 LET C=1
425 PRINT AT 2,0;" "
426 NEXT I
427 GOTO 428
428 LET C=1
429 PRINT AT 2,0;" "
430 NEXT I
431 GOTO 432
432 LET C=1
433 PRINT AT 2,0;" "
434 NEXT I
435 GOTO 436
436 LET C=1
437 PRINT AT 2,0;" "
438 NEXT I
439 GOTO 440
440 LET C=1
441 PRINT AT 2,0;" "
442 NEXT I
443 GOTO 444
444 LET C=1
445 PRINT AT 2,0;" "
446 NEXT I
447 GOTO 448
448 LET C=1
449 PRINT AT 2,0;" "
450 NEXT I
451 GOTO 452
452 LET C=1
453 PRINT AT 2,0;" "
454 NEXT I
455 GOTO 456
456 LET C=1
457 PRINT AT 2,0;" "
458 NEXT I
459 GOTO 460
460 LET C=1
461 PRINT AT 2,0;" "
462 NEXT I
463 GOTO 464
464 LET C=1
465 PRINT AT 2,0;" "
466 NEXT I
467 GOTO 468
468 LET C=1
469 PRINT AT 2,0;" "
470 NEXT I
471 GOTO 472
472 LET C=1
473 PRINT AT 2,0;" "
474 NEXT I
475 GOTO 476
476 LET C=1
477 PRINT AT 2,0;" "
478 NEXT I
479 GOTO 480
480 LET C=1
481 PRINT AT 2,0;" "
482 NEXT I
483 GOTO 484
484 LET C=1
485 PRINT AT 2,0;" "
486 NEXT I
487 GOTO 488
488 LET C=1
489 PRINT AT 2,0;" "
490 NEXT I
491 GOTO 492
492 LET C=1
493 PRINT AT 2,0;" "
494 NEXT I
495 GOTO 496
496 LET C=1
497 PRINT AT 2,0;" "
498 NEXT I
499 GOTO 500
500 LET C=1
501 PRINT AT 2,0;" "
502 NEXT I
503 GOTO 504
504 LET C=1
505 PRINT AT 2,0;" "
506 NEXT I
507 GOTO 508
508 LET C=1
509 PRINT AT 2,0;" "
510 NEXT I
511 GOTO 512
512 LET C=1
513 PRINT AT 2,0;" "
514 NEXT I
515 GOTO 516
516 LET C=1
517 PRINT AT 2,0;" "
518 NEXT I
519 GOTO 520
520 LET C=1
521 PRINT AT 2,0;" "
522 NEXT I
523 GOTO 524
524 LET C=1
525 PRINT AT 2,0;" "
526 NEXT I
527 GOTO 528
528 LET C=1
529 PRINT AT 2,0;" "
530 NEXT I
531 GOTO 532
532 LET C=1
533 PRINT AT 2,0;" "
534 NEXT I
535 GOTO 536
536 LET C=1
537 PRINT AT 2,0;" "
538 NEXT I
539 GOTO 540
540 LET C=1
541 PRINT AT 2,0;" "
542 NEXT I
543 GOTO 544
544 LET C=1
545 PRINT AT 2,0;" "
546 NEXT I
547 GOTO 548
548 LET C=1
549 PRINT AT 2,0;" "
550 NEXT I
551 GOTO 552
552 LET C=1
553 PRINT AT 2,0;" "
554 NEXT I
555 GOTO 556
556 LET C=1
557 PRINT AT 2,0;" "
558 NEXT I
559 GOTO 560
560 LET C=1
561 PRINT AT 2,0;" "
562 NEXT I
563 GOTO 564
564 LET C=1
565 PRINT AT 2,0;" "
566 NEXT I
567 GOTO 568
568 LET C=1
569 PRINT AT 2,0;" "
570 NEXT I
571 GOTO 572
572 LET C=1
573 PRINT AT 2,0;" "
574 NEXT I
575 GOTO 576
576 LET C=1
577 PRINT AT 2,0;" "
578 NEXT I
579 GOTO 580
580 LET C=1
581 PRINT AT 2,0;" "
582 NEXT I
583 GOTO 584
584 LET C=1
585 PRINT AT 2,0;" "
586 NEXT I
587 GOTO 588
588 LET C=1
589 PRINT AT 2,0;" "
590 NEXT I
591 GOTO 592
592 LET C=1
593 PRINT AT 2,0;" "
594 NEXT I
595 GOTO 596
596 LET C=1
597 PRINT AT 2,0;" "
598 NEXT I
599 GOTO 600
600 LET C=1
601 PRINT AT 2,0;" "
602 NEXT I
603 GOTO 604
604 LET C=1
605 PRINT AT 2,0;" "
606 NEXT I
607 GOTO 608
608 LET C=1
609 PRINT AT 2,0;" "
610 NEXT I
611 GOTO 612
612 LET C=1
613 PRINT AT 2,0;" "
614 NEXT I
615 GOTO 616
616 LET C=1
617 PRINT AT 2,0;" "
618 NEXT I
619 GOTO 620
620 LET C=1
621 PRINT AT 2,0;" "
622 NEXT I
623 GOTO 624
624 LET C=1
625 PRINT AT 2,0;" "
626 NEXT I
627 GOTO 628
628 LET C=1
629 PRINT AT 2,0;" "
630 NEXT I
631 GOTO 632
632 LET C=1
633 PRINT AT 2,0;" "
634 NEXT I
635 GOTO 636
636 LET C=1
637 PRINT AT 2,0;" "
638 NEXT I
639 GOTO 640
640 LET C=1
641 PRINT AT 2,0;" "
642 NEXT I
643 GOTO 644
644 LET C=1
645 PRINT AT 2,0;" "
646 NEXT I
647 GOTO 648
648 LET C=1
649 PRINT AT 2,0;" "
650 NEXT I
651 GOTO 652
652 LET C=1
653 PRINT AT 2,0;" "
654 NEXT I
655 GOTO 656
656 LET C=1
657 PRINT AT 2,0;" "
658 NEXT I
659 GOTO 660
660 LET C=1
661 PRINT AT 2,0;" "
662 NEXT I
663 GOTO 664
664 LET C=1
665 PRINT AT 2,0;" "
666 NEXT I
667 GOTO 668
668 LET C=1
669 PRINT AT 2,0;" "
670 NEXT I
671 GOTO 672
672 LET C=1
673 PRINT AT 2,0;" "
674 NEXT I
675 GOTO 676
676 LET C=1
677 PRINT AT 2,0;" "
678 NEXT I
679 GOTO 680
680 LET C=1
681 PRINT AT 2,0;" "
682 NEXT I
683 GOTO 684
684 LET C=1
685 PRINT AT 2,0;" "
686 NEXT I
687 GOTO 688
688 LET C=1
689 PRINT AT 2,0;" "
690 NEXT I
691 GOTO 692
692 LET C=1
693 PRINT AT 2,0;" "
694 NEXT I
695 GOTO 696
696 LET C=1
697 PRINT AT 2,0;" "
698 NEXT I
699 GOTO 700
700 LET C=1
701 PRINT AT 2,0;" "
702 NEXT I
703 GOTO 704
704 LET C=1
705 PRINT AT 2,0;" "
706 NEXT I
707 GOTO 708
708 LET C=1
709 PRINT AT 2,0;" "
710 NEXT I
711 GOTO 712
712 LET C=1
713 PRINT AT 2,0;" "
714 NEXT I
715 GOTO 716
716 LET C=1
717 PRINT AT 2,0;" "
718 NEXT I
719 GOTO 720
720 LET C=1
721 PRINT AT 2,0;" "
722 NEXT I
723 GOTO 724
724 LET C=1
725 PRINT AT 2,0;" "
726 NEXT I
727 GOTO 728
728 LET C=1
729 PRINT AT 2,0;" "
730 NEXT I
731 GOTO 732
732 LET C=1
733 PRINT AT 2,0;" "
734 NEXT I
735 GOTO 736
736 LET C=1
737 PRINT AT 2,0;" "
738 NEXT I
739 GOTO 740
740 LET C=1
741 PRINT AT 2,0;" "
742 NEXT I
743 GOTO 744
744 LET C=1
745 PRINT AT 2,0;" "
746 NEXT I
747 GOTO 748
748 LET C=1
749 PRINT AT 2,0;" "
750 NEXT I
751 GOTO 752
752 LET C=1
753 PRINT AT 2,0;" "
754 NEXT I
755 GOTO 756
756 LET C=1
757 PRINT AT 2,0;" "
758 NEXT I
759 GOTO 760
760 LET C=1
761 PRINT AT 2,0;" "
762 NEXT I
763 GOTO 764
764 LET C=1
765 PRINT AT 2,0;" "
766 NEXT I
767 GOTO 768
768 LET C=1
769 PRINT AT 2,0;" "
770 NEXT I
771 GOTO 772
772 LET C=1
773 PRINT AT 2,0;" "
774 NEXT I
775 GOTO 776
776 LET C=1
777 PRINT AT 2,0;" "
778 NEXT I
779 GOTO 780
780 LET C=1
781 PRINT AT 2,0;" "
782 NEXT I
783 GOTO 784
784 LET C=1
785 PRINT AT 2,0;" "
786 NEXT I
787 GOTO 788
788 LET C=1
789 PRINT AT 2,0;" "
790 NEXT I
791 GOTO 792
792 LET C=1
793 PRINT AT 2,0;" "
794 NEXT I
795 GOTO 796
796 LET C=1
797 PRINT AT 2,0;" "
798 NEXT I
799 GOTO 800
800 LET C=1
801 PRINT AT 2,0;" "
802 NEXT I
803 GOTO 804
804 LET C=1
805 PRINT AT 2,0;" "
806 NEXT I
807 GOTO 808
808 LET C=1
809 PRINT AT 2,0;" "
810 NEXT I
811 GOTO 812
812 LET C=1
813 PRINT AT 2,0;" "
814 NEXT I
815 GOTO 816
816 LET C=1
817 PRINT AT 2,0;" "
818 NEXT I
819 GOTO 820
820 LET C=1
821 PRINT AT 2,0;" "
822 NEXT I
823 GOTO 824
824 LET C=1
825 PRINT AT 2,0;" "
826 NEXT I
827 GOTO 828
828 LET C=1
829 PRINT AT 2,0;" "
830 NEXT I
831 GOTO 832
832 LET C=1
833 PRINT AT 2,0;" "
834 NEXT I
835 GOTO 836
836 LET C=1
837 PRINT AT 2,0;" "
838 NEXT I
839 GOTO 840
840 LET C=1
841 PRINT AT 2,0;" "
842 NEXT I
843 GOTO 844
844 LET C=1
845 PRINT AT 2,0;" "
846 NEXT I
847 GOTO 848
848 LET C=1
849 PRINT AT 2,0;" "
850 NEXT I
851 GOTO 852
852 LET C=1
853 PRINT AT 2,0;" "
854 NEXT I
855 GOTO 856
856 LET C=1
857 PRINT AT 2,0;" "
858 NEXT I
859 GOTO 860
860 LET C=1
861 PRINT AT 2,0;" "
862 NEXT I
863 GOTO 864
864 LET C=1
865 PRINT AT 2,0;" "
866 NEXT I
867 GOTO 868
868 LET C=1
869 PRINT AT 2,0;" "
870 NEXT I
871 GOTO 872
872 LET C=1
873 PRINT AT 2,0;" "
874 NEXT I
875 GOTO 876
876 LET C=1
877 PRINT AT 2,0;" "
878 NEXT I
879 GOTO 880
880 LET C=1
881 PRINT AT 2,0;" "
882 NEXT I
883 GOTO 884
884 LET C=1
885 PRINT AT 2,0;" "
886 NEXT I
887 GOTO 888
888 LET C=1
889 PRINT AT 2,0;" "
890 NEXT I
891 GOTO 892
892 LET C=1
893 PRINT AT 2,0;" "
894 NEXT I
895 GOTO 896
896 LET C=1
897 PRINT AT 2,0;" "
898 NEXT I
899 GOTO 900
900 LET C=1
901 PRINT AT 2,0;" "
902 NEXT I
903 GOTO 904
904 LET C=1
905 PRINT AT 2,0;" "
906 NEXT I
907 GOTO 908
908 LET C=1
909 PRINT AT 2,0;" "
910 NEXT I
911 GOTO 912
912 LET C=1
913 PRINT AT 2,0;" "
914 NEXT I
915 GOTO 916
916 LET C=1
917 PRINT AT 2,0;" "
918 NEXT I
919 GOTO 920
920 LET C=1
921 PRINT AT 2,0;" "
922 NEXT I
923 GOTO 924
924 LET C=1
925 PRINT AT 2,0;" "
926 NEXT I
927 GOTO 928
928 LET C=1
929 PRINT AT 2,0;" "
930 NEXT I
931 GOTO 932
932 LET C=1
933 PRINT AT 2,0;" "
934 NEXT I
935 GOTO 936
936 LET C=1
937 PRINT AT 2,0;" "
938 NEXT I
939 GOTO 940
940 LET C=1
941 PRINT AT 2,0;" "
942 NEXT I
943 GOTO 944
944 LET C=1
945 PRINT AT 2,0;" "
946 NEXT I
947 GOTO 948
948 LET C=1
949 PRINT AT 2,0;" "
950 NEXT I
951 GOTO 952
952 LET C=1
953 PRINT AT 2,0;" "
954 NEXT I
955 GOTO 956
956 LET C=1
957 PRINT AT 2,0;" "
958 NEXT I
959 GOTO 960
960 LET C=1
961 PRINT AT 2,0;" "
962 NEXT I
963 GOTO 964
964 LET C=1
965 PRINT AT 2,0;" "
966 NEXT I
967 GOTO 968
968 LET C=1
969 PRINT AT 2,0;" "
970 NEXT I
971 GOTO 972
972 LET C=1
973 PRINT AT 2,0;" "
974 NEXT I
975 GOTO 976
976 LET C=1
977 PRINT AT 2,0;" "
978 NEXT I
979 GOTO 980
980 LET C=1
981 PRINT AT 2,0;" "
982 NEXT I
983 GOTO 984
984 LET C=1
985 PRINT AT 2,0;" "
986 NEXT I
987 GOTO 988
988 LET C=1
989 PRINT AT 2,0;" "
990 NEXT I
991 GOTO 992
992 LET C=1
993 PRINT AT 2,0;" "
994 NEXT I
995 GOTO 996
996 LET C=1
997 PRINT AT 2,0;" "
998 NEXT I
999 GOTO 1000
1000 LET C=1
1001 PRINT AT 2,0;" "
1002 NEXT I
1003 GOTO 1004
1004 LET C=1
1005 PRINT AT 2,0;" "
1006 NEXT I
1007 GOTO 1008
1008 LET C=1
1009 PRINT AT 2,0;" "
1010 NEXT I
1011 GOTO 1012
1012 LET C=1
1013 PRINT AT 2,0;" "
1014 NEXT I
1015 GOTO 1016
1016 LET C=1
1017 PRINT AT 2,0;" "
1018 NEXT I
1019 GOTO 1020
1020 LET C=1
1021 PRINT AT 2,0;" "
1022 NEXT I
1023 GOTO 1024
1024 LET C=1
1025 PRINT AT 2,0;" "
1026 NEXT I
1027 GOTO 1028
1028 LET C=1
1029 PRINT AT 2,0;" "
1030 NEXT I
1031 GOTO 1032
1032 LET C=1
1033 PRINT AT 2,0;" "
1034 NEXT I
1035 GOTO 1036
1036 LET C=1
1037 PRINT AT 2,0;" "
1038 NEXT I
1039 GOTO 1040
1040 LET C=1
1041 PRINT AT 2,0;" "
1042 NEXT I
1043 GOTO 1044
1044 LET C=1
1045 PRINT AT 2,0;" "
1046 NEXT I
1047 GOTO 1048
1048 LET C=1
1049 PRINT AT 2,0;" "
1050 NEXT I
1051 GOTO 1052
1052 LET C=1
1053 PRINT AT 2,0;" "
1054 NEXT I
1055 GOTO 1056
1056 LET C=1
1057 PRINT AT 2,0;" "
1058 NEXT I
1059 GOTO 1060
1060 LET C=1
1061 PRINT AT 2,0;" "
1062 NEXT I
1063 GOTO 1064
1064 LET C=1
1065 PRINT AT 2,0;" "
1066 NEXT I
1067 GOTO 1068
1068 LET C=1
1069 PRINT AT 2,0;" "
1070 NEXT I
1071 GOTO 1072
1072 LET C=1
1073 PRINT AT 2,0;" "
1074 NEXT I
1075 GOTO 1076
1076 LET C=1
1077 PRINT AT 2,0;" "
1078 NEXT I
1079 GOTO 1080
1080 LET C=1
1081 PRINT AT 2,0;" "
1082 NEXT I
1083 GOTO 1084
1084 LET C=1
1085 PRINT AT 2,0;" "
1086 NEXT I
1087 GOTO 1088
1088 LET C=1
1089 PRINT AT 2,0;" "
1090 NEXT I
1091 GOTO 1092
1092 LET C=1
1093 PRINT AT 2,0;" "
1094 NEXT I
1095 GOTO 1096
1096 LET C=1
1097 PRINT AT 2,0;" "
1098 NEXT I
1099 GOTO 1100
1100 LET C=1
1101 PRINT AT 2,0;" "
1102 NEXT I
1103 GOTO 1104
1104 LET C=1
1105 PRINT AT 2,0;" "
1106 NEXT I
1107 GOTO 1108
1108 LET C=1
1109 PRINT AT 2,0;" "
1110 NEXT I
1111 GOTO 1112
1112 LET C=1
1113 PRINT AT 2,0;" "
1114 NEXT I
1115 GOTO 1116
1116 LET C=1
1117 PRINT AT 2,0;" "
1118 NEXT I
1119 GOTO 1120
1120 LET C=1
1121 PRINT AT 2,0;" "
1122 NEXT I
1123 GOTO 1124
1124 LET C=1
1125 PRINT AT 2,0;" "
1126 NEXT I
1127 GOTO 1128
1128 LET C=1
1129 PRINT AT 2,0;" "
1130 NEXT I
1131 GOTO 1132
1132 LET C=1
1133 PRINT AT 2,0;" "
1134 NEXT I
1135 GOTO 1136
1136 LET C=1
1137 PRINT AT 2,0;" "
1138 NEXT I
1139 GOTO 1140
1140 LET C=1
1141 PRINT AT 2,0;" "
1142 NEXT I
1143 GOTO 1144
1144 LET C=1
1145 PRINT AT 2,0;" "
1146 NEXT I
1147 GOTO 1148
1148 LET C=1
1149 PRINT AT 2,0;" "
1150 NEXT I
1151 GOTO 1152
1152 LET C=1
1153 PRINT AT 2,0;" "
1154 NEXT I
1155 GOTO 1156
1156 LET C=1
1157 PRINT AT 2,0;" "
1158 NEXT I
1159 GOTO 1160
1160 LET C=1
1161 PRINT AT 2,0;" "
1162 NEXT I
1163 GOTO 1164
1164 LET C=1
1165 PRINT AT 2,0;" "
1166 NEXT I
1167 GOTO 1168
1168 LET C=1
1169 PRINT AT 2,0;" "
1170 NEXT I
1171 GOTO 1172
1172 LET C=1
1173 PRINT AT 2,0;" "
1174 NEXT I
1175 GOTO 1176
1176 LET C=1
1177 PRINT AT 2,0;" "
1178 NEXT I
1179 GOTO 1180
1180 LET C=1
1181 PRINT AT 2,0;" "
1182 NEXT I
1183 GOTO 1184
1184 LET C=1
1185 PRINT AT 2,0;" "
1186 NEXT I
1187 GOTO 1188
1188 LET C=1
1189 PRINT AT 2,0;" "
1190 NEXT I
1191 GOTO 1192
1192 LET C=1
1193 PRINT AT 2,0;" "
1194 NEXT I
1195 GOTO 1196
1196 LET C=1
1197 PRINT AT 2,0;" "
1198 NEXT I
1199 GOTO 1200
1200 LET C=1
1201 PRINT AT 2,0;" "
1202 NEXT I
1203 GOTO 1204
1204 LET C=1
1205 PRINT AT 2,0;" "
1206 NEXT I
1207 GOTO 1208
1208 LET C=1
1209 PRINT AT 2,0;" "
1210 NEXT I
1211 GOTO 1212
1212 LET C=1
1213 PRINT AT 2,0;" "
1214 NEXT I
1215 GOTO 1216
1216 LET C=1
1217 PRINT AT 2,0;" "
1218 NEXT I
1219 GOTO 1220
1220 LET C=1
1221 PRINT AT 2,0;" "
1222 NEXT I
1223 GOTO 1224
1224 LET C=1
1225 PRINT AT 2,0;" "
1226 NEXT I
1227 GOTO 1228
1228 LET C=1
1229 PRINT AT 2,0;" "
1230 NEXT I
1231 GOTO 1232
1232 LET C=1
1233 PRINT AT 2,0;" "
1234 NEXT I
1235 GOTO 1236
1236 LET C=1
1237 PRINT AT 2,0;" "
1238 NEXT I
1239 GOTO 1240
1240 LET C=1
1241 PRINT AT 2,0;" "
1242 NEXT I
1243 GOTO 1244
1244 LET C=1
1245 PRINT AT 2,0;" "
1246 NEXT I
1247 GOTO 1248
1248 LET C=1
1249 PRINT AT 2,0;" "
1250 NEXT I
1251 GOTO 1252
1252 LET C=1
1253 PRINT AT 2,0;" "
1254 NEXT I
1255 GOTO 1256
1256 LET C=1
1257 PRINT AT 2,0;" "
1258 NEXT I
1259 GOTO 1260
1260 LET C=1
1261 PRINT AT 2,0;" "
1262 NEXT I
1263 GOTO 1264
1264 LET C=1
1265 PRINT AT 2,0;" "
1266 NEXT I
1267 GOTO 1268
1268 LET C=1
1269 PRINT AT 2,0;" "
1270 NEXT I
1271 GOTO 1272
1272 LET C=1
1273 PRINT AT 2,0;" "
1274 NEXT I
1275 GOTO 1276
1276 LET C=1
1277 PRINT AT 2,0;" "
1278 NEXT I
1279 GOTO 1280
1280 LET C=1
1281 PRINT AT 2,0;" "
1282 NEXT I
1283 GOTO 1284
1284 LET C=1
1285 PRINT AT 2,0;" "
1286 NEXT I
1287 GOTO 1288
1288 LET C=1
1289 PRINT AT 2,0;" "
1290 NEXT I
1291 GOTO 1292
1292 LET C=1
1293 PRINT AT 2,0;" "
1294 NEXT I
1295 GOTO 1296
1296 LET C=1
1297 PRINT AT 2,0;" "
1298 NEXT I
1299 GOTO 1300
1300 LET C=1
1301 PRINT AT 2,0;" "
1302 NEXT I
1303 GOTO 1304
1304 LET C=1
1305 PRINT AT 2,0;" "
1306 NEXT I
1307 GOTO 1308
1308 LET C=1
1309 PRINT AT 2,0;" "
1310 NEXT I
1311 GOTO 1312
1312 LET C=1
1313 PRINT AT 2,0;" "
1314 NEXT I
1315 GOTO 1316
1316 LET C=1
1317 PRINT AT 2,0;" "
1318 NEXT I
1319 GOTO 1320
1320 LET C=1
1321 PRINT AT 2,0;" "
1322 NEXT I
1323 GOTO 1324
1324 LET C=1
1325 PRINT AT 2,0;" "
1326 NEXT I
1327 GOTO 1328
1328 LET C=1
1329 PRINT AT 2,0;" "
1330 NEXT I
1331 GOTO 1332
1332 LET C=1
1333 PRINT AT 2,0;" "
1334 NEXT I
1335 GOTO 1336
1336 LET C=1
1337 PRINT AT 2,0;" "
1338 NEXT I
1339 GOTO 1340
1340 LET C=1
1341 PRINT AT 2,0;" "
1342 NEXT I
1343 GOTO 1344
1344 LET C=1
1345 PRINT AT 2,0;" "
1346 NEXT I
1347 GOTO 1348
1348 LET C=1
13
```

MUNCHER II

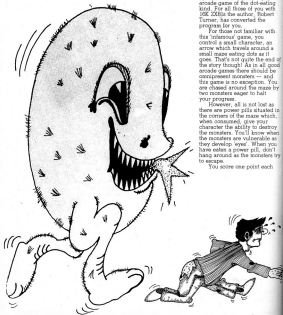
Monsters will try to catch you while you're quietly going dotty!

In the June/July issue of *ZX Computing*, 1983, appeared the program *Muncher* — a Spectrum version of the popular arcade game of the dot-eating kind. For all those of you with MSX, ZX81s the author, Robert Turner, has converted the program for you.

For those not familiar with this infamous game, you control a small character, an arrow which travels around a small maze eating dots as it goes. That's not quite the end of the story though! As in all good arcade games there should be omnipresent monsters — and this game is no exception. You are chased around the maze by two monsters eager to halt your progress.

However, all is not lost as there are power pills situated in the corners of the maze which, when consumed, give your character the ability to destroy the monsters. You'll know when the monsters are vulnerable as they develop 'eyes'. When you have eaten a power pill, don't hang around as the monsters try to escape.

You score one point each



time you eat a dot, and power pills and monsters are worth 10 points. Every row and then a 'X' sign will appear under the monster's den and this too is worth 10 points if consumed.

VARIABLES

The variables used in the program, Muncher II, are as follows:

S — Score
HS — High Score
T — The score when the screen has been cleared.
B\$ — The variable which stores the maze.
COUNT — If this variable is less than 30 the monsters can be eaten. If it is more than 30 the

monsters can eat your character.
LIVES — The number of lives you have left.
Y and X — The position of your character.



AS — The shape of your character.
GX and GY — The position of the first monster.
GX1 and GY1 — The position of the second monster.
C — This variable controls the movement of the monster. If C = -1 then the monster moves away from your character, if C=1 then the monster chases you.
DS — The shape under the first monster.
ES — The shape under the second monster.
A — The skill level.
N — Allocated for general use.

So, don't sit there — type in the program and watch out for the monsters!

```

1 LET S=0
2 LET HS=0
3 LET T=1000
4 DIM B$(255,255)
5 LET COUNT=30
6 LET DS=0
7 GOSUB 7000
8 LET DSOCR=1
9 LET S=0
10 LET H=1
11 LET B$(11)=
20 LET B$(21)=
30 LET B$(31)=
40 LET B$(41)=
50 LET B$(51)=
60 LET B$(61)=
70 LET B$(71)=
80 LET B$(81)=
90 LET B$(91)=
100 LET B$(101)=
110 LET B$(111)=
120 LET B$(121)=
130 LET B$(131)=
140 LET B$(141)=
150 LET B$(151)=
160 LET B$(161)=
170 LET B$(171)=
180 LET B$(181)=
190 LET B$(191)=
200 LET B$(201)=
210 LET B$(211)=
220 FOR H=1 TO 31
230 PRINT AT H,1;DSOCR
240 NEXT H
250 LET Y=14

```

```

260 LET B$=""
270 LET X=15
280 LET B$=""
290 LET B$=""
300 LET B$=""
310 LET B$=""
320 LET B$=""
330 LET B$=""
340 LET B$=""
350 LET B$=""
360 LET B$=""
370 LET B$=""
380 LET B$=""
390 LET B$=""
400 LET B$=""
410 LET B$=""
420 LET B$=""
430 LET B$=""
440 LET B$=""
450 LET B$=""
460 LET B$=""
470 LET B$=""
480 LET B$=""
490 LET B$=""
500 LET B$=""
510 LET B$=""
520 LET B$=""
530 LET B$=""
540 LET B$=""
550 LET B$=""
560 LET B$=""
570 LET B$=""
580 LET B$=""
590 LET B$=""
600 LET B$=""
610 LET B$=""
620 LET B$=""
630 LET B$=""
640 LET B$=""
650 LET B$=""
660 LET B$=""
670 LET B$=""
680 LET B$=""
690 LET B$=""
700 LET B$=""
710 LET B$=""
720 LET B$=""
730 LET B$=""
740 LET B$=""
750 LET B$=""
760 LET B$=""
770 LET B$=""
780 LET B$=""
790 LET B$=""
800 LET B$=""
810 LET B$=""
820 LET B$=""
830 LET B$=""
840 LET B$=""
850 LET B$=""
860 LET B$=""
870 LET B$=""
880 LET B$=""
890 LET B$=""
900 LET B$=""
910 LET B$=""
920 LET B$=""
930 LET B$=""
940 LET B$=""
950 LET B$=""
960 LET B$=""
970 LET B$=""
980 LET B$=""
990 LET B$=""

```


THE RIGHT CHORD



Having difficulty getting your fingers round those tricky guitar chords? Bring you ZX81 into play.



When RUN, this program draws a simulation of a guitar neck and asks the user to input a number corresponding to the particular chord you wish to see on the screen.

GETTING IN TUNE...

The program asks the user to input a number from one to ten and then the chord corresponding to the number will be displayed with the relevant finger positions. The chord name is displayed in a box beside the diagram and the program then waits for a further input.

You could add two more chords to the published program by changing line 82 to read:

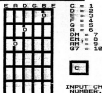
```
82 IF C<1 OR C>12 THEN
  GOTO 80
```

and enter the co-ordinates for the new chords in the subroutines at lines 9900 and 9910. Then, you would have to alter the menu in line 7065. The chords included in the program as it stands are C, D, E, F, G, A, Dm, Em, Am, and G7.

Now if only someone could write a program to tune the guitar...

SAMPLE SCREEN.

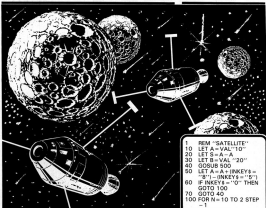
GUITAR CHORD FINDER.



INPUT CHORD
NUMBER.

ENEMY SATELLITE

Don't let the enemy get your secrets.



Written for the unexpanded Z801, the object of the game is to shoot the satellites out of the sky before they can relay information on your military bases back to the enemy.

As the enemy satellites traverse across the top of the screen, it is your job to move the ground to air missile launcher until you can get a good shot at the enemy device. Movement of the missile launcher is achieved by pressing the 'B' key to go left and the 'R' key to move right. Key 'Q' is used to launch a missile at the satellite.

Should a satellite reach the

far left-hand side of the screen, it is assumed that the device has escaped destruction and has thus relayed all of the secret data on your military establishments. This is achieved in line 530 with the 'Q', and an error 3 message denotes the end of the game.

To hit a satellite, you must aim for the 'Q' character of the device. If you do manage to destroy the satellite, line 160 causes an explosion to be displayed: this is followed by a PAUSE statement causing the screen to blink (just to add a touch of realism). When the game finishes, your score is displayed.

```

1  REM "SATELLITE"
10  LET A=VAL"10"
20  LET S=A-A
30  LET B=VAL"30"
40  GOSUB 500
50  LET A=A+(INKEY$="B")-(INKEY$="R")
60  IF INKEY$="Q" THEN
    GOTO 100
70  GOTO 40
100 FOR N=10 TO 2 STEP
    -1
110 GOSUB 500
120 PRINT AT N,A+2;" * "
    AT N+1,A+2;" "
130 NEXT N
140 IF A=INT B THEN
    GOTO 180
150 GOTO 40
160 PRINT AT 2,B;">X<"
170 PAUSE 50
180 LET S=S+B
190 CLS
200 GOTO 30
500 LET B=B-.25
510 PRINT AT 2,INT B;"
    =0=";"AT 11,A;"
    "■"■"■"
520 IF S<0.2 THEN PRINT
    AT 2,0;"GAME OVER
    -";INT S;G
530 RETURN
  
```

ON TARGET

Two great programs — Target and Missile Launcher.

When RUN, Target displays a target and a randomly placed star near the centre of the screen. You must then, using the 'N' key to move left and the 'M' key to move right, position the target so that the star is in the middle of it. Pressing the 'X' key will then fire at the star.

Confirmation of a hit on the star is given by the star turning to greenish video. The program ends if you fail to hit a star.

You set the difficulty of the game pointed at the beginning of the game; if you input a value below 15, the game is very hard indeed! The difficulty is calculated in the FOR...TO loop in line 50.

MISSILE LAUNCHER

In this program, you control a ground-based missile launcher with which you can fire missiles at oncoming alien attackers.

You can move your missile launcher using the 'N' key to go right and the 'M' key to go left. The 'Z' key is used to launch the missile. You can cheat by holding the 'Z' key down all the time as you move, but this will show up at the end of the game when the number of shots you took is displayed.

The game can be altered so that the missile rises at a faster rate. This is done by changing line 150 to read:

```
150 IF B<= 15 THEN LET
  B=B-2
```

If you make this change, you should also alter line 70 to read:

```
70 IF B<= 0 THEN LET
  B=20
```



The listing for Missile Launcher.

```
10 LET A=0
15 LET X=15
20 LET Y=0
30 LET V=0
40 LET S=0
50 GOINT AT 0,X:Y=0
60 GOINT AT 0,1:2=0
70 IF 0=0 THEN LET S=0
80 GOINT AT Y,2=1
90 IF 2=1
100 IF 2=1 THEN LET Y=Y+5
110 IF 2=1 THEN LET Y=0
120 IF INKEY$="N" THEN LET X=X-1
130 IF INKEY$="M" THEN LET X=X+1
140 IF INKEY$="Z" THEN LET S=15
150 IF INKEY$="Z" THEN LET S=1
160 IF 0=15 THEN LET S=0-1
170 IF Y=0 AND 2=X THEN GOTO 3
180 IF X=2 AND 0=Y THEN GOTO 3
190 CLS
200 GOTO 20
210 PRINT AT 15,0,"ALIENS HAVE
  LANDED"
220 STOP
230 PRINT AT 15,0," YOU HIT TH
  E ALIEN IN ".S," SHOTS"
```

A sample screen dump
from the program,
Missile Launcher.



The listing for Target.

```
5 PRINT "INPUT DIFFICULTY"
10 INPUT D
15 CLS
20 LET S=0
30 LET X=15
40 LET Y=15
50 LET V=INT (RND*20)
60 FOR T=1 TO D
70 GOINT AT X,Y: I =INT X*Y
80 IF X=0,Y=0 THEN I=INT X*Y
90 LET Y=Y+(INKEY$="N")-INKEY$
100 IF INKEY$="X" AND Y=0:1 T
  THEN GOTO 150
110 PRINT AT 15,0,"a"
120 GOINT AT 0,15,"YOU HIT "S
130 STOP
140 PRINT AT 15,0,"E"
150 LET S=S+1
160 PAUSE 50
170 LET S=1
180 GOTO 20
```

YOU HIT 5

A sample screen dump
from the program, Target.



"Run more than ten tasks on a ZX81-FORTH ROM?"

Sure! More than 10 tasks simultaneously and, in some cases, up to 360 times faster! That's what replacing the basic ROM with the new FORTH does for the ZX81 - and more!

The brain behind the breakthrough belongs to David Husband, another building for positive software on the strength of it. Already orders are flooding in and it's easy to see why.

The ZX81-FORTH ROM gives you a totally new system. In addition to multi-tasking and split screen window capability, you can also edit a program while there are four others are executing, schedule tasks to run from 50 times a second to once a year, and with a further modification switch between FORTH and BASIC, whenever you like.

The ZX81-FORTH ROM gives you an eternal keyboard with a 64 character buffer and repeat - it supports the 16k, 32k, 64k RAM packs, it's a 4k FORTH compatible and it supports the ZX printer.

The price, too, is almost unbelievable. As a "fit in yourself" (compact) with manual, it's just £12.95 + post. Add £2 plus UK VAT (Europe, £10 outside Europe) and send your order to the address below.

Skyway
SOFTWARE
David Husband
73 Cumnor Road, Bicester, Oxon,
OX4 1JN, UK
Tel: 02555 281381
International +44 2555 281381

~~50 games for the Sinclair Spectrum, 16K/32K.~~
~~There are 100 super high quality games on one tape!~~

STOP PRESS:

There are now sixty games on this fantastic compilation, yes 60 super high resolution arcade games that you will have hours and hours of enjoyment from, guaranteed!!

60 of the best games around including: Pacman, Invaders, Frogger, Fruit Machines, Defender, Lunar Lander, Pongoon, plus much more!! All on one tape for only £3.95.

50 games for the ZX81 - £5.99. An amazing simulation of the side-winder arcade fruit machine, features include random holds and nudges, flip-winding reels to a winning line. Visually stunning graphics to 'state of the art' machine code, true arcade sound effects.
Introductory price £5.50.

ONLY AVAILABLE BY MAIL ORDER FROM

ANCO SOFTWARE (DEPT ASP)
25 Cornwell Street, Crowthorne,
Reading RG4 7JF

ANCO SOFTWARE ... THE ULTIMATE EXPERIENCE

It's easy to complain about an advertisement. Once you know how.

One of the ways we keep a check on the advertising that appears in the press, on posters and in the cinema is by responding to consumers' complaints.

Any complaint sent to us is considered carefully and, if there's a case to answer, a full investigation is made.

If you think you've got good reason to complain about an advertisement, send off for a copy of our free leaflet.

It will tell you all you need to know to help us process your complaint as quickly as possible. ✓

The Advertising Standards Authority. Most advertisements are wrong, we've here to put it right.

ASA Ltd, Dept 1 Brook House,
Torrington Place, London WC1E 7HN

The space donated in the interests of high standards of advertising.

FREE **LEAFLET** **HOW** **TO** **COMPLAIN**

Copy it in 5 minutes
No postage
The leaflet
Explains
How to
Complain
To
The
ASA
The
Advertising
Standards
Authority

IT'S THE WAY TO
GET YOUR
COMPLAINT
HANDLED
FAST

THEY SAY THE WORLD WILL
ONE DAY BE ONE
OF THE
BEST
PLACES
TO LIVE
IN

SEND FOR YOUR COPY
To
Advertising Standards
Authority
Dept 1
Brook House
Torrington Place
London WC1E 7HN

2-99!!!

SALES
HIGHLIGHTS
1995

SMAUG'S LAIR

Can you destroy the dragon's lair?

You are the Gurner Captain on a mission to destroy the dragon's castle.

You have five cannons and 12 cannonballs at your disposal . . . but . . . beware . . . the sound of your guns will wake the dragon. When he flies, time stands still and although he has poor eyesight, he will breathe fire and try to destroy all of your weapons. If he does . . . *run for your life!*

To dislodge the most bricks, hit the castle as high as you can by carefully choosing how much gunpowder to use and what angle to set the cannon at. If 6,000 bricks fall then the castle is in ruins and Smaug must seek a new wormhole. Good luck!!!

CANNON AND BALL

When you first RUN this program you are asked if you require instructions in the usual way. If you reply 'yes' (Y) you are given a brief introduction to the program to get the adrenalin up.

Then, the 'Battle Status' screen is shown and is returned to throughout the game. You are shown the current scores, and the castle will be demolished in proportion to the number of bricks you have knocked out. At the top of the screen, you see how far the computer has set the cannon from the castle ready for your first shot (this is random between 500 and 2,000 metres). You are then invited to enter the size of the gunpowder charge (25 to 35lbs) and what angle of elevation you wish the cannon to be set to (0 to 89 degrees).

Should the cannonball reach its target (the castle is 400

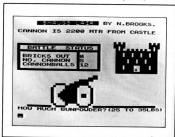
metres high) then the Y co-ordinate of the cannonball plot determines how many bricks are knocked out. (See program lines 1071 to 1079.) When in flight, the dragon may observe the ball but it will always re-appear in the same position after the dragon has passed — unless, of course, it burns the cannonball to a cinder! You are finally asked if you wish to continue the game. Answering 'no' (N), puts you in retreat and the dragon wins.

At the end of the game (either with 6,000 bricks out of the castle or no cannonballs left) there is an appropriate 'picture show' to watch which I leave you to discover.

The odds slightly favour the Gurner Captain once an appreciation of charge and angle has been gained. Bear in mind that an angle of 45 degrees will send the ball the maximum distance for any

choice of charge (at which, the castle is seldom damaged incidentally) and an angle of say 60 degrees will result in the same final range as that of 30 degrees, so each 15 degrees either side of 45 degrees. However, the castle presents a smaller target when dropped on from above rather than hit broadside on, so for a more difficult game try always choosing angles over 45 degrees. I have yet to win doing this!

The more gunpowder you use, the faster the cannonball leaves the cannon and hence the further the cannonball will fly. Similarly, the angle of elevation affects the range. (I have used Newton's formulae for the notion of projection from my old 'A' level maths notes for this part of the program and have consulted books on cannons for details of range and charge to make the scientific aspects of the program as accurate as possible).



A graphical display of the camera being adjusted to the angle chosen is shown next. (I make no apology for the slowness of this part of the program as I wanted to introduce the concept of 'angles' to my young family. Adding lines 3305 FAST and 3310 SLOW will alleviate any boredom.)

INTO BATTLE

The main action comes on the 'Battled' screen which follows on from the previous screen automatically. The cannon is displayed at the computer-selected distance from the castle and the cannonball is fired. Every fifth plot of the cannonball, the dragon flies from the castle. Unbeknown to the player, it has chosen at random whether to attack the cannon or the cannonball. If the dragon chooses the cannon there is a 50% chance of a hit. If the cannonball is selected, then the odds of a hit are only 20%.

VARIABLES ON A THEME

Listed here are the main variables used throughout the review.

BRICK	The number of bricks knocked out of the castle (8,000 bricks knocked out will win the game for you).
BALL	The number of cannonballs remaining (zero cannonballs left will result in you losing the game).
GUNS	The number of cannon you have remaining (zero cannon left will result in you losing the game).
FOUT	The 'I give up' code.
M	The distance between the cannon and the castle.
N	The random number used to generate the variable, M.
G	If G is equal to one, the dragon attacks the cannonball only.
X and Y	The co-ordinates of the cannonball.
A	The input for the angle of the cannon in radians.
B	The angle converted into radians.
U	The input for the amount (in lbs) of gunpowder used.

Other variables are used but they only exist to aid the running of the program.

[illegible]

```

0000 PLOT 12.5/80-143-3-4-H, Y, 500, 10
0001 IF X<10 THEN UNPLOT 12.5/80
0002 -12.5/80, Y, 1500, -5
0003 IF X<2000 THEN X/2500 AND X:
0004 THEN GOTO 0000
0005 GOTO 0100
0006 GOTO 0100
0007 IF X<10 AND X<10+1500 AND Y
0008 THEN GOTO 1070
0009 NEXT X
0010 PRINT AT 12.5, "CANNONBALL"
0011 GOTO 1100
0012 GOTO 1100
0013 PRINT AT 10.25, "SMITZ!!" AT
0014 10.25, "*****" AT 17.25, "*****"
0015
0016 IF Y<400 AND Y<500 THEN LE
0017 GOTO-500-1250
0018 IF Y<500 AND Y<600 THEN LE
0019 GOTO-500-1200
0020 IF Y<600 AND Y<700 THEN LE
0021 GOTO-500-1150
0022 IF Y<700 AND Y<800 THEN LE
0023 GOTO-500-1100
0024 IF Y<800 THEN LE GOTO-500
0025 PRINT AT 10.2, "BALL STRUCK
0026 AT Y, X, RETARD UP!"
0027 GOTO 1100
0028 PRINT AT 12.5, "CANNONBALL
0029 SHOT!"
0030 LE BALL-BALL-1
0031 PRINT AT 20.8, "DO YOU WANT
0032 COFFEE?"
0033 IF ANSWER="Y" THEN GOTO 1100
0034 IF ANSWER="N" THEN LE ROUTE
0035
0036 GOTO 00
0037 MEN "GUNNER"
0038 CLS
0039 PRINT "GUNNER SERGEANT...LO
0040 C, ORANGE OF "O, LES AND SET
0041 Cannon To "A, DEGREE"
0042 COLOR 2000
0043 PRINT
0044 OVER
0045

```

[illegible]

[illegible][illegible][illegible][illegible]

```

0136 IF S=0 GOTO THEN GOTO 0100
0138 LET S=S+1
0140 LET E=INT (1000/(S-1000)) /200
0142 LET P=INT (1000-(S-1000)) /200
0144 FOR R=0 TO E
0146 PRINT AT S-R,E;E;
0148 NEXT R
0150 FOR S=0 TO F
0152 PRINT AT S+E,S+E+1;
0154 NEXT S
0156 PRINT AT 10,0;
0158 PRINT AT 0,10;0;0;
0160 PRINT AT 0,10;0;0;
0170 PRINT AT 10,10;0;0;
0200 IF S=0 GOTO 0200 THEN GOTO 710
0300 IF S=0=0 THEN GOTO 7000
0400 IF S=0=0 THEN GOTO 7000
0500 IF S=0=1 THEN GOTO 7000
0600 PRINT AT 0,0;CANNON IS "
0700 WITH PRESS CASTLE
0800 PRINT AT 0,0; "HOW MUCH GUN
0900 CONVERTED TO CASTLE"
1000 INPUT M
1100 IF U=0 OR U>25 THEN GOTO 0
1200 PRINT AT 0,0; "ENTER CANNON
1300 WHILE INK=0 GOTO
1400 RETURN
1500 REM *****INSTRUCTIONS*****
1600 PRINT AT 0,0; "YOU ARE THE D
1700 CANNON ON A
1800 "YOUR MISSION TO DESTROY T
1900 PRINT "CASTLE."
2000 PRINT "YOU HAVE 5 CANNON AN
2100 0 10 CANNON"
2200 PRINT "BALLS AT YOUR DISPOS
2300 "BUT...
2400 "BEWARE...THE SOUND O
2500 "YOUR GUNS
2600 PRINT "WILL MAKE THE CANNON
2700 PRINT "WHEN HE FLIES TIME S
2800 PRINT "AND ALTHOUGH OF BOO
2900 PRINT "HE WILL BREATHE FIRE
3000 PRINT "DESTROY ALL OF YOUR
3100 PRINT "HE SAYS...RUN FOR YO
3200 PRINT "IF
3300 PRINT "TO DILLODO MOST BR
3400 PRINT "CASTLE AS HIGH AS YO
3500 PRINT "CAREFULLY CHOOSING H
3600 PRINT "POWDER TO USE AND UN
3700 PRINT "SET THE CANNON AT IF
3800 PRINT "FALL THEN THE CASTLE
3900 PRINT "AND SHOTS MUST SEEK
4000 PRINT "A NEW HOLE.
4100 PRINT "***** LUCK *****
4200 PRINT "
4300 PRINT "
4400 PRINT "
4500 PRINT "
4600 PRINT "
4700 PRINT "
4800 PRINT "
4900 PRINT "
5000 PRINT "
5100 PRINT "
5200 PRINT "
5300 PRINT "
5400 PRINT "
5500 PRINT "
5600 PRINT "
5700 PRINT "
5800 PRINT "
5900 PRINT "
6000 PRINT "
6100 PRINT "
6200 PRINT "
6300 PRINT "
6400 PRINT "
6500 PRINT "
6600 PRINT "
6700 PRINT "
6800 PRINT "
6900 PRINT "
7000 PRINT "
7100 PRINT "
7200 PRINT "
7300 PRINT "
7400 PRINT "
7500 PRINT "
7600 PRINT "
7700 PRINT "
7800 PRINT "
7900 PRINT "
8000 PRINT "
8100 PRINT "
8200 PRINT "
8300 PRINT "
8400 PRINT "
8500 PRINT "
8600 PRINT "
8700 PRINT "
8800 PRINT "
8900 PRINT "
9000 PRINT "
9100 PRINT "
9200 PRINT "
9300 PRINT "
9400 PRINT "
9500 PRINT "
9600 PRINT "
9700 PRINT "
9800 PRINT "
9900 PRINT "

```

NAVAL MANOEUVRES

Can you stay afloat in this ZX81 version of the popular board game?



Written for the 16K ZX81, this version of 'Battleships' involves you taking on the computer in a naval conflict.

IN THE SWIM

As in the game we all know and love, the idea of the game is to sink your opponent's fleet before it manages to sink yours.

When you first run the program, the computer first provides you with a few brief instructions on how to play the

game. When you are sure you understand the workings of the game, the computer sets up its fleet on the right-hand playing area. Once this has been done, you position your own fleet.

You do this under instruction from the computer which invites you to give the co-ordinates of each ship in your fleet. You must position one battleship (represented by four squares), two cruisers (three squares), three destroyers (two squares) and four submarines (one square each). You position your fleet by first inputting the row co-ordinate on your 10 by

10 playing grid on the left-hand part of the screen.

You begin the game by having three shots at the computer's fleet. This is done in a similar manner to how you inputted your fleet, in that the computer asks you first for the row co-ordinate followed by the column co-ordinate. The computer will then have three shots at your fleet and so on. All shots and hits are shown clearly on the screen, and the winner is the one who destroys the other's fleet first. The player has the same chance of winning as the computer, so have fun!

[illegible][illegible]

1. The first step in the process is to identify the problem or issue that needs to be addressed. This involves gathering information and understanding the context of the problem.

```

8743 IF A=8 THEN GOTO 8888
8744 IF B=8 THEN GOTO 8888
8745 IF A=8 THEN GOTO 8708
8746 FOR I=1 TO 3
8747 IF A=8,B=1-1:1:1:1 THEN GO
8748 TO 8708
8749 NEXT I
8750 FOR I=1 TO 8
8751 LET A=8,B=1-1:1:1:1="C"
8752 NEXT I
8753 GOTO 8888
8754 FOR I=1 TO 3
8755 NEXT I
8756 FOR I=1 TO 3
8757 LET A=8,B=1-1:1:1:1="C"
8758 NEXT I
8759 GOTO 8888
8760 IF A=8 THEN GOTO 8888
8761 FOR I=1 TO 3
8762 IF A=8,B=1-1:1:1:1:1 THEN GO
8763 TO 8708
8764 NEXT I
8765 FOR I=1 TO 3
8766 LET A=8,B=1-1:1:1:1="C"
8767 NEXT I
8768 GOTO 8888
8769 IF A=8 THEN GOTO 8888
8770 FOR I=1 TO 3
8771 IF A=8,B=1-1:1:1:1:1 THEN GO
8772 TO 8708
8773 NEXT I
8774 FOR I=1 TO 3
8775 LET A=8,B=1-1:1:1:1="C"
8776 NEXT I
8777 GOTO 8888
8778 IF A=8 THEN GOTO 8888
8779 FOR I=1 TO 3
8780 IF A=8,B=1-1:1:1:1:1 THEN GO
8781 TO 8708
8782 NEXT I
8783 FOR I=1 TO 3
8784 LET A=8,B=1-1:1:1:1="C"
8785 NEXT I
8786 GOTO 8888
8787 IF A=8 THEN GOTO 8888
8788 FOR I=1 TO 3
8789 IF A=8,B=1-1:1:1:1:1 THEN GO
8790 TO 8708
8791 NEXT I
8792 FOR I=1 TO 3
8793 LET A=8,B=1-1:1:1:1="C"
8794 NEXT I
8795 GOTO 8888
8796 IF A=8 THEN GOTO 8888
8797 FOR I=1 TO 3
8798 IF A=8,B=1-1:1:1:1:1 THEN GO
8799 TO 8708
8800 NEXT I
8801 FOR I=1 TO 3
8802 LET A=8,B=1-1:1:1:1="C"
8803 NEXT I
8804 GOTO 8888
8805 IF A=8 THEN GOTO 8888
8806 FOR I=1 TO 3
8807 IF A=8,B=1-1:1:1:1:1 THEN GO
8808 TO 8708
8809 NEXT I
8810 FOR I=1 TO 3
8811 LET A=8,B=1-1:1:1:1="C"
8812 NEXT I
8813 GOTO 8888
8814 IF A=8 THEN GOTO 8888
8815 FOR I=1 TO 3
8816 IF A=8,B=1-1:1:1:1:1 THEN GO
8817 TO 8708
8818 NEXT I
8819 FOR I=1 TO 3
8820 LET A=8,B=1-1:1:1:1="C"
8821 NEXT I
8822 GOTO 8888
8823 IF A=8 THEN GOTO 8888
8824 FOR I=1 TO 3
8825 IF A=8,B=1-1:1:1:1:1 THEN GO
8826 TO 8708
8827 NEXT I
8828 FOR I=1 TO 3
8829 LET A=8,B=1-1:1:1:1="C"
8830 NEXT I
8831 GOTO 8888
8832 IF A=8 THEN GOTO 8888
8833 FOR I=1 TO 3
8834 IF A=8,B=1-1:1:1:1:1 THEN GO
8835 TO 8708
8836 NEXT I
8837 FOR I=1 TO 3
8838 LET A=8,B=1-1:1:1:1="C"
8839 NEXT I
8840 GOTO 8888
8841 IF A=8 THEN GOTO 8888
8842 FOR I=1 TO 3
8843 IF A=8,B=1-1:1:1:1:1 THEN GO
8844 TO 8708
8845 NEXT I
8846 FOR I=1 TO 3
8847 LET A=8,B=1-1:1:1:1="C"
8848 NEXT I
8849 GOTO 8888
8850 IF A=8 THEN GOTO 8888
8851 FOR I=1 TO 3
8852 IF A=8,B=1-1:1:1:1:1 THEN GO
8853 TO 8708
8854 NEXT I
8855 FOR I=1 TO 3
8856 LET A=8,B=1-1:1:1:1="C"
8857 NEXT I
8858 GOTO 8888
8859 IF A=8 THEN GOTO 8888
8860 FOR I=1 TO 3
8861 IF A=8,B=1-1:1:1:1:1 THEN GO
8862 TO 8708
8863 NEXT I
8864 FOR I=1 TO 3
8865 LET A=8,B=1-1:1:1:1="C"
8866 NEXT I
8867 GOTO 8888
8868 IF A=8 THEN GOTO 8888
8869 FOR I=1 TO 3
8870 IF A=8,B=1-1:1:1:1:1 THEN GO
8871 TO 8708
8872 NEXT I
8873 FOR I=1 TO 3
8874 LET A=8,B=1-1:1:1:1="C"
8875 NEXT I
8876 GOTO 8888
8877 IF A=8 THEN GOTO 8888
8878 FOR I=1 TO 3
8879 IF A=8,B=1-1:1:1:1:1 THEN GO
8880 TO 8708
8881 NEXT I
8882 FOR I=1 TO 3
8883 LET A=8,B=1-1:1:1:1="C"
8884 NEXT I
8885 GOTO 8888
8886 IF A=8 THEN GOTO 8888
8887 FOR I=1 TO 3
8888 IF A=8,B=1-1:1:1:1:1 THEN GO
8889 TO 8708
8890 NEXT I
8891 FOR I=1 TO 3
8892 LET A=8,B=1-1:1:1:1="C"
8893 NEXT I
8894 GOTO 8888
8895 IF A=8 THEN GOTO 8888
8896 FOR I=1 TO 3
8897 IF A=8,B=1-1:1:1:1:1 THEN GO
8898 TO 8708
8899 NEXT I
8900 FOR I=1 TO 3
8901 LET A=8,B=1-1:1:1:1="C"
8902 NEXT I
8903 GOTO 8888
8904 IF A=8 THEN GOTO 8888
8905 FOR I=1 TO 3
8906 IF A=8,B=1-1:1:1:1:1 THEN GO
8907 TO 8708
8908 NEXT I
8909 FOR I=1 TO 3
8910 LET A=8,B=1-1:1:1:1="C"
8911 NEXT I
8912 GOTO 8888
8913 IF A=8 THEN GOTO 8888
8914 FOR I=1 TO 3
8915 IF A=8,B=1-1:1:1:1:1 THEN GO
8916 TO 8708
8917 NEXT I
8918 FOR I=1 TO 3
8919 LET A=8,B=1-1:1:1:1="C"
8920 NEXT I
8921 GOTO 8888
8922 IF A=8 THEN GOTO 8888
8923 FOR I=1 TO 3
8924 IF A=8,B=1-1:1:1:1:1 THEN GO
8925 TO 8708
8926 NEXT I
8927 FOR I=1 TO 3
8928 LET A=8,B=1-1:1:1:1="C"
8929 NEXT I
8930 GOTO 8888
8931 IF A=8 THEN GOTO 8888
8932 FOR I=1 TO 3
8933 IF A=8,B=1-1:1:1:1:1 THEN GO
8934 TO 8708
8935 NEXT I
8936 FOR I=1 TO 3
8937 LET A=8,B=1-1:1:1:1="C"
8938 NEXT I
8939 GOTO 8888
8940 IF A=8 THEN GOTO 8888
8941 FOR I=1 TO 3
8942 IF A=8,B=1-1:1:1:1:1 THEN GO
8943 TO 8708
8944 NEXT I
8945 FOR I=1 TO 3
8946 LET A=8,B=1-1:1:1:1="C"
8947 NEXT I
8948 GOTO 8888
8949 IF A=8 THEN GOTO 8888
8950 FOR I=1 TO 3
8951 IF A=8,B=1-1:1:1:1:1 THEN GO
8952 TO 8708
8953 NEXT I
8954 FOR I=1 TO 3
8955 LET A=8,B=1-1:1:1:1="C"
8956 NEXT I
8957 GOTO 8888
8958 IF A=8 THEN GOTO 8888
8959 FOR I=1 TO 3
8960 IF A=8,B=1-1:1:1:1:1 THEN GO
8961 TO 8708
8962 NEXT I
8963 FOR I=1 TO 3
8964 LET A=8,B=1-1:1:1:1="C"
8965 NEXT I
8966 GOTO 8888
8967 IF A=8 THEN GOTO 8888
8968 FOR I=1 TO 3
8969 IF A=8,B=1-1:1:1:1:1 THEN GO
8970 TO 8708
8971 NEXT I
8972 FOR I=1 TO 3
8973 LET A=8,B=1-1:1:1:1="C"
8974 NEXT I
8975 GOTO 8888
8976 IF A=8 THEN GOTO 8888
8977 FOR I=1 TO 3
8978 IF A=8,B=1-1:1:1:1:1 THEN GO
8979 TO 8708
8980 NEXT I
8981 FOR I=1 TO 3
8982 LET A=8,B=1-1:1:1:1="C"
8983 NEXT I
8984 GOTO 8888
8985 IF A=8 THEN GOTO 8888
8986 FOR I=1 TO 3
8987 IF A=8,B=1-1:1:1:1:1 THEN GO
8988 TO 8708
8989 NEXT I
8990 FOR I=1 TO 3
8991 LET A=8,B=1-1:1:1:1="C"
8992 NEXT I
8993 GOTO 8888
8994 IF A=8 THEN GOTO 8888
8995 FOR I=1 TO 3
8996 IF A=8,B=1-1:1:1:1:1 THEN GO
8997 TO 8708
8998 NEXT I
8999 FOR I=1 TO 3
9000 LET A=8,B=1-1:1:1:1="C"
9001 NEXT I
9002 GOTO 8888
9003 IF A=8 THEN GOTO 8888
9004 FOR I=1 TO 3
9005 IF A=8,B=1-1:1:1:1:1 THEN GO
9006 TO 8708
9007 NEXT I
9008 FOR I=1 TO 3
9009 LET A=8,B=1-1:1:1:1="C"
9010 NEXT I
9011 GOTO 8888
9012 IF A=8 THEN GOTO 8888
9013 FOR I=1 TO 3
9014 IF A=8,B=1-1:1:1:1:1 THEN GO
9015 TO 8708
9016 NEXT I
9017 FOR I=1 TO 3
9018 LET A=8,B=1-1:1:1:1="C"
9019 NEXT I
9020 GOTO 8888
9021 IF A=8 THEN GOTO 8888
9022 FOR I=1 TO 3
9023 IF A=8,B=1-1:1:1:1:1 THEN GO
9024 TO 8708
9025 NEXT I
9026 FOR I=1 TO 3
9027 LET A=8,B=1-1:1:1:1="C"
9028 NEXT I
9029 GOTO 8888
9030 IF A=8 THEN GOTO 8888
9031 FOR I=1 TO 3
9032 IF A=8,B=1-1:1:1:1:1 THEN GO
9033 TO 8708
9034 NEXT I
9035 FOR I=1 TO 3
9036 LET A=8,B=1-1:1:1:1="C"
9037 NEXT I

```

HOUSE OF HORROR

I was a dark and stormy night (as Victorian writer, Edward Bulwer Lytton, used to say) when Chris Cox decided to write this program!

The plot of the game is that you have been transported to a deserted mansion, full of hidden treasure. All of the treasure was kindly left by the last owner of the house, but unfortunately he also left a number of his 'pets' to guard his fortune. And yes, you've guessed it, his 'pets' (in the shape of monsters!) are some too pleased by your intrusion. And if that wasn't

enough, you have to watch out for the roving reaper who has a liking for people's heads — it's not too interested in the rest of your body though (niced I say more!).

GO EXPLORING

Instructions appear throughout the program telling you in which direction you can go. You move around the randomly-generated selection of halls and stairways constantly exploring until you find the

Dare you enter the haunted house?

treasure, keys to open some of the doors and, of course, a grim selection of monsters.

The more coins you find, the less frequent large finds become; however, as you come across less coins, you'll find more keys and have more encounters which means you can explore more of the house.

The game ends when you have found over 2,500 coins, run out of strength wading off the monsters or you've had your head removed by the roving reaper!

Happy hunting!

```

10 LET H=0
20 LET S=200
30 LET C=0
40 LET R=0
50 LET H=0
60 PRINT AT 3.0;"THE HOUSE"
70 PRINT AT 3.4;"INSTRUCTIONS FOLLOW"
80 PAUSE 200
90 CLS
100 PRINT AT 2.10;"THE HOUSE."
110 PRINT TAB 10;"-----"
120 PRINT
130 PRINT "YOU HAVE BEEN TRANSPORTED TO"
140 PRINT "A DESERTED MANSION, A LILY GARDEN IT"
150 PRINT "TREASURE IS HIDDEN, IT WAS LEFT"
160 PRINT "BY THE LAST OWNER,"
170 PRINT
180 PRINT "AS WELL AS HIS THREE MONSTER PETS"
190 PRINT "HIS PET MONSTERS TO GUARD IT."
200 PRINT "BEWARE THE CLUTCHES OF THE REAPER!"
210 PRINT "THE REAPER....."
220 PRINT AT 21.0;"PRESS ANY KEY TO START."
230 PAUSE 3000
240 CLS
250 LET R4="A HALL"
260 LET R5="A DOOR"
270 LET R6="A STAIRWAY"
280 LET R=INT (RND*3)+1
290 IF R=1 THEN LET X5=C5
300 IF R=2 THEN LET X5=B5
310 IF R=3 THEN LET X5=B5
320 IF R=3 THEN LET X5=B5
330 PRINT AT 21.0;"YOU ARE FACING"
340 CLS
350 SCROLL
360 IF X5=B5 THEN GOSUB 400
370 IF X5=B5 THEN GOSUB 400
380 IF X5=C5 THEN GOSUB 500
390 SCROLL
400 SCROLL
410 PRINT TAB 11;MOVE (2);EXPLORE
420 STATUS
430 INPUT 0
440 IF R=1 THEN GOSUB 500
450 IF R=2 THEN GOSUB 640
460 IF R=3 THEN GOSUB 710
470 RETURN
480 PRINT "(1) OPEN (2) LEAVE"
490 INPUT 2
500 IF 2=1 THEN GOSUB 670
510 IF 2=2 THEN RETURN
520 RETURN
530 PRINT "(1) UP (2) DOWN (3) LEFT (4) RIGHT"
540 INPUT 0
550 IF 0=1 THEN GOSUB 810
560 IF 0=2 THEN GOSUB 880
570 IF 0=3 THEN RETURN
580 RETURN
590 SCROLL
600 SCROLL
610 PRINT "(1) NORTH (2) SOUTH (3) EAST (4) WEST"
620 INPUT X
630 RETURN
640 SCROLL
650 SCROLL
660 LET A=INT (RND*3)+1
670 IF A=1 THEN GOSUB 1000
680 IF A=1 THEN GOSUB 1070
690 IF A=2 THEN GOSUB 1100
700 RETURN
710 CLS
720 PRINT AT 3.10;"STATUS"
730 PRINT
740 PRINT
750 PRINT TAB 4;"STRENGTH"
760 PRINT
770 PRINT TAB 4;"COINS"
780 PRINT
790 PRINT TAB 4;"KEYS"
800 PRINT
810 PRINT TAB 4;"MONSTERS DEAD"
820 PRINT
830 PRINT AT 15.0;"SCORE :";C5
840 IF C5<1000 THEN C5=1100

```



```

800 PRINT AT 21.0;"PRESS ANY KE
V TO CONTINUE....."
810 PAUSE 3000
820 CLS
830 RETURN
870 LET X=INT (RND*4)+1
880 IF X=3 THEN GOSUB 1000
890 IF X=8 THEN GOSUB 1000
900 RETURN
910 LET X=INT (RND*2)+1
920 SCROLL
930 SCROLL
940 IF X=3 THEN PRINT "O.K. YOU
ARE UP A FLOOR"
950 IF X=2 THEN PRINT "NO WAY U
P THERE."
960 IF RND<.7 THEN GOSUB 1000
970 RETURN
980 LET R=INT (RND*2)+1
990 SCROLL
1000 SCROLL
1010 IF R=3 THEN PRINT "O.K. YOU
ARE DOWN A FLOOR"
1020 IF R=2 THEN PRINT "NO WAY D
OWN THERE."
1030 IF RND<.7 THEN GOSUB 1000
1040 SCROLL
1050 SCROLL
1060 LET R=INT (RND*30)+1
1070 IF R=3 THEN LET H$=" A TOOL
"
1080 IF R=3 THEN LET H$=" A DRESS
"
1090 IF R=4 THEN LET H$=" A VAMP
"
1100 IF R=5 THEN LET H$=" A WERE
WOLF"
1110 IF R=6 THEN LET H$=" A HYDRA
"
1120 IF R=7 THEN LET H$=" A SHIL
"
1130 IF R=8 THEN LET H$=" A HEDU
"
1140 IF R=9 THEN LET H$=" THE RE
"
1150 PRINT "YOUVE MET";H$
1160 SCROLL
1170 SCROLL
1180 PRINT "(1) ATTACK (2) RETRE
AT"
1190 INPUT P
1210 IF P=3 AND R=3 THEN RETURN
1220 IF P=1 THEN GOTO 1260
1230 SCROLL
1240 SCROLL
1250 PRINT "HE DOES NOT LIKE YOU
"
1260 LET R=INT (RND*99)+1
1270 SCROLL
1280 SCROLL
1290 PRINT "HE HAS A PUNCH FACTO
R OF"
1300 SCROLL
1310 SCROLL
1320 PRINT "INPUT PUNCH FACTOR (
1-9)"
1330 INPUT P
1340 LET PP=P*P
1350 LET QQ=INT (PP/2)
1360 IF P=1 THEN GOTO 2100
1370 IF PP=R THEN GOTO 1390
1380 IF PP=R THEN GOTO 1390
1390 LET R=R-PP
1400 GOTO 1070
1410 SCROLL
1420 SCROLL
1430 PRINT "YOU WON.STRENGTH IS
NOW"
1440 LET H=R+1
1450 RETURN
1460 SCROLL
1470 SCROLL
1480 LET R=INT (RND*500)+50
1490 PRINT "YOUVE FOUND A R.";"OO
"
1500 LET C=C+1
1510 LET S=S+1
1520 IF C=500 THEN GOTO 1600
1530 RETURN
1540 SCROLL
1550 SCROLL
1560 PRINT "YOUVE FOUND A KEY."
1570 LET R=R+1
1580 LET S=S+20
1590 RETURN
1600 SCROLL
1610 SCROLL
1620 PRINT "DOOR LOCKED.GOT A KE
Y?"
1630 IF R=3 THEN RETURN
1640 SCROLL
1650 SCROLL
1660 PRINT "YES."
1670 LET R=R-1
1680 SCROLL
1690 SCROLL
1700 PRINT "THE DOOR HAS OPENED."
1710 LET R=INT (RND*4)+1
1720 IF R=2 THEN GOSUB 1400
1730 IF R=3 THEN GOSUB 1400
1740 IF R=3 ON R+3 THEN GOSUB 14
00
1750 RETURN
1760 PRINT AT 2.10;"THE END."
1770 CLS
1780 PRINT TAB 10;"-----"
1790 PRINT AT 2.8;" YOU MET YO
U AND
1800 IN THE HANDS."
1810 PRINT "OF THE REAPER.HE CUT
"
1820 PRINT "OFF."
1830 PRINT
1840 PRINT " BEFORE YOU DIED Y
OU COULD"
1850 PRINT "(1)2+3+C+(R+5)+(H+1)
"
1860 PRINT
1870 PRINT " YOU FOUND "C;" " C
"
1880 PRINT
1890 PRINT TAB 10;"WELL DONE."
1900 GOTO 2000
1910 STOP
1920 STOP
1930 CLS
1940 PRINT AT 2.10;"THE HOUSE."
1950 PRINT TAB 10;"-----"
1960 PRINT
1970 PRINT " YOUVE SENTEN THE
HOUSE AND"
1980 PRINT "ITS HORRORS."
1990 PRINT
2000 PRINT " WELL DONE.YOU FOU
ND " C
"
2010 PRINT "GOING DURING THE GRI
E AND RELE
"
2020 PRINT
2030 PRINT K;" KEYS LEFT.YOU AL
O KILLED"
2040 PRINT
2050 PRINT H;" MONSTERS."
2060 PRINT
2070 PRINT
2080 IF R+(12+30+C+(R+5)+(H+1))+
THEN LET R=12+30+C+(R+5)+(H+1)
2090 PRINT TAB 6;"HIGHEST SCORE"
2100 PRINT TAB 10;"
"
2110 PRINT AT 21.0;"ANOTHER GO (
0-9)"
2120 INPUT R$
2130 IF R$="R" THEN STOP
2140 CLS
2150 GOTO 20
2160 CLS
2170 PRINT AT 2.10;"THE HOUSE."
2180 PRINT TAB 10;"-----"
2190 PRINT
2200 PRINT "YOUVE RUN OUT OF STR
"
2210 PRINT
2220 PRINT "THE HOUSE HAS WON AG
AIN."
2230 GOTO 2000

```

DROP OUT

A ZX81 version of the popular arcade game.

This is a game for the ZX81 with at least 32K of RAM and has been written so that, apart from the enjoyment I hope you'll get from the game, you will also learn more about the capabilities of the computer.

The idea of the game is to drop your arrow, a 'V', into the mouth of a bottle. If you miss the mouth of the bottle or land in one of the other bottles, you score the amount of points indicated at the bottom of the bottle (printed in inverse print). You have ten attempts to score as many points as possible, and you lose one of your goes for each 'V' dropped. To drop a 'V', press any key apart from Break or Shift.

GOTTA LOTTA BOTTLE

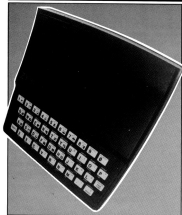
When the program is RUN, the bottles appear and your 'V' moves rapidly across the screen. Pressing a key will make your arrow fall from its path, hopefully into the mouth of a bottle. When your ten attempts have been exhausted, your score and the highest score so far will be displayed.

The program uses the memory mapped display of the ZX81 with at least 32K. I have deliberately not included instructions in my program so that users with more than 1K but less than 12K can also enjoy the game.

I have used variables to represent areas of the program, for example:

70 GOSUB DRAWG

This line makes the program go



PROGRAM VARIABLES

- | | |
|---------|---|
| 5 | — Initialise the main variables used in the program. |
| 10 | — Finds the beginning of the display file. |
| 30 | — The starting position of your arrow character. |
| 210-230 | — The routine to find out if your arrow has hit anything. |
| 260 | — The routine to print on the screen how much you scored. |
| 710 | — Prints your points on the screen. |

MATHS MAZE

A great game with an educational twist.



At the start of the game you must select a level of difficulty, from one to five, five being the hardest and almost impossible. The object of the game is to move around the maze, using the inverse quotes, using the arrow keys, 'S' to 'E', until you find one of the signs: '+', '-', 'x' or '/'. When you find one of these signs, you simply 'eat' it. The computer

will then clear the screen, and set you five questions, their hardness depending on the level of difficulty selected at the start of the game. After these five questions, you are given a 'percentage eating', depending on how many signs you answered correctly. After this, press any key and the faithful 2281 will put you back in the maze to carry on the game.

ON LEAVE

You may leave the maze at any time, but obviously the idea is to get as many points as possible. You get one point for each correctly answered addition or subtraction question, and two points for each multiplication or divide question.


```

0000 NEXT X
0010 RETURN
0020 REM *****
0030 LET A=10
0040 LET B=3
0050 LET P=PEEK 10000+200*PEEK 1
0060 *1
0070 LET COR=2
0080 LET SCORE=0
0090 RETURN
0100 REM *****
0110 PRINT AT 4,8," YOU IN "
1100 IN 0 MAKE AND YOUR TASK
1200 PRINT TO THE EXIT
1300 PRINT " YOU UP "
1400 PICK UP THE STICKS
1500 AND CHECKER THE CORRESPONDING CU
1600 POINT " TO HAVE USE
1700 *
0180 POINT "PLEASE ENTER LEV
EL OF DIFFICULTY"
0190 PRINT
0200 REM *****
0210 LET A=INKEY$
0220 IF A="1" THEN GOTO 0400
0230 IF A="2" THEN GOTO 0500
0240 IF A="3" THEN GOTO 0600
0250 IF A="4" THEN GOTO 0700
0260 GOTO 0300
0270 FOR X=1 TO 10
0280 PRINT AT X,0:
0290 NEXT X
0300 LET DIFF=VAL A
0310 RETURN
0320 REM *****
0330 REM *****

```



1000

1000

1	2	3	4
---	---	---	---

1000

2008	7	1	2008	7	1
2008	7	1	2008	7	1
2008	7	1	2008	7	1
2008	7	1	2008	7	1



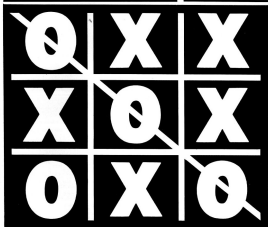
YOU GOT A RIGHT,
THAT TA AS DEFENSE.

***** "N.E." TO CONTINUE *****

[illegible]

NAUGHT BUT A GAME

The classic game of naughts and crosses for your ZX81.



This program has the ZX81 playing a pretty mean game of Naughts and Crosses against a willing opponent. The program requires about 7K of RAM to operate.

In this game you are the 'naughts' and the computer is the 'crosses'. The program is divided into several parts:

- DRAWC** — Draws the computer's piece.
- DRAWP** — Draws the player's piece.
- WIN** — Checks to see if someone has won or

COMP

if it is a draw.

- The computer's move.

PLAYER

- The player's move.

CLEAR

- Clears the player's prompts.

GRID

- Prints the playing grid.

Each part of the program is called by the GOSUB command, for example GOSUB COMP is the subroutine for the computer's moves.

THREE IN A ROW

The following is a brief guide to

some of the more important lines in the program:

- 5005-5030** Find if one of the players has a row of three counters. The player gets two points for each win. The ZX81 gets five points for each win.
- 5320** Work out the average value of RND.
- 5330** Ensure the print position is in the top-left hand corner.
- 6005-6008** Make sure that the scores do not run onto the board.
- 6009**
- 6020-6210**

[illegible]


```

8000 PRINT AT 10,0;"IT IS A DRAW"
8010 US BOTH SET:AT 20,0;"I POINT"
8020 PRINT AT 10,0;"YOU POINT"
8030 PRINT AT 20,0;"I POINT"
8040 GOTO 8000
8050 LET COMPU=COMPU+1
8060 LET HUMAN=HUMAN+1
8070 GOSUB CLEAR
8080 PRINT AT 10,0;"PRESS C TO"
8090 PRINT AT 20,0;"SPACE BAR TO"
8100 GOTO 8000
8110 IF INKEY="C" THEN GOTO 8000
8120 GOTO 8100
8130 FOR Q=1 TO 3
8140 FOR T=1 TO 3
8150 LET X10,Q,T=0
8160 NEXT T
8170 NEXT Q
8180 LET HUMAN=HUMAN+2
8190 FOR T=1 TO 100
8200 PRINT AT 10,0;"YOU WIN 2 PO"
8210 IF T=10,0;"IT IS A DRAW"
8220 NEXT T
8230 GOSUB CLEAR
8240 GOTO 8070
8250 REM
8260 REM
8270 REM
8280 REM
8290 REM
8300 FOR Q=1 TO 11
8310 LET X10,Q,T=0
8320 NEXT Q
8330 LET HUMAN=HUMAN+2
8340 NEXT T
8350 IF X10,Q,T=0 THEN GOTO 7000
8360 FOR T=1 TO 3
8370 FOR Q=1 TO 3
8380 IF X10,Q,T=0 THEN GOTO 8000
8390 NEXT Q
8400 NEXT T
8410 RETURN
8420 IF Q=1 THEN LET X10,Q,T=1
8430 IF Q=2 THEN LET X10,Q,T=2
8440 IF Q=3 THEN LET X10,Q,T=3
8450 IF T=1 THEN LET Y10,Q,T=1
8460 IF T=2 THEN LET Y10,Q,T=2
8470 IF T=3 THEN LET Y10,Q,T=3
8480 IF X10,Q,T=0 THEN GOTO 8000
8490 LET X10,Q,T=0
8500 GOTO 8000
8510 RETURN
8520 FOR T=1 TO 3
8530 FOR Q=1 TO 3
8540 LET XC=INT (RND*3)+1
8550 LET YC=INT (RND*3)+1
8560 IF YC=0 OR XC=0 OR YC=3 OR
8570 XC=3 THEN GOTO 7000
8580 IF X10,Q,T=0 THEN GOTO 8000
8590 NEXT Q
8600 NEXT T
8610 GOTO 8000
8620 REM
8630 REM
8640 REM
8650 PRINT AT 0,0;
8660 PRINT
8670 PRINT "EX-01"
8680 PRINT
8690 PRINT "00000"
8700 PRINT
8710 PRINT "YOU"
8720 PRINT
8730 PRINT
8740 PRINT "00000"

```

```

8000 PRINT
8010 PRINT
8020 PRINT
8030 PRINT
8040 PRINT
8050 PRINT
8060 PRINT
8070 PRINT
8080 PRINT
8090 PRINT
8100 PRINT
8110 PRINT
8120 LET H1=7-LEN (STR$ COMPU)
8130 LET H2=7-LEN (STR$ HUMAN)
8140 PRINT AT 0,H1,COMPU,AT 7,H2
8150 HUMAN
8160 RETURN
8170 REM
8180 REM
8190 REM
8200 REM
8210 REM
8220 REM
8230 REM
8240 REM
8250 REM
8260 REM
8270 REM
8280 REM
8290 REM
8300 REM
8310 REM
8320 REM
8330 REM
8340 REM
8350 REM
8360 REM
8370 REM
8380 REM
8390 REM
8400 REM
8410 REM
8420 REM
8430 REM
8440 REM
8450 REM
8460 REM
8470 REM
8480 REM
8490 REM
8500 REM
8510 REM
8520 REM
8530 REM
8540 REM
8550 REM
8560 REM
8570 REM
8580 REM
8590 REM
8600 REM
8610 REM
8620 REM
8630 REM
8640 REM
8650 REM
8660 REM
8670 REM
8680 REM
8690 REM
8700 REM
8710 REM
8720 REM
8730 REM
8740 REM
8750 REM
8760 REM
8770 REM
8780 REM
8790 REM
8800 REM
8810 REM
8820 REM
8830 REM
8840 REM
8850 REM
8860 REM
8870 REM
8880 REM
8890 REM
8900 REM
8910 REM
8920 REM
8930 REM
8940 REM
8950 REM
8960 REM
8970 REM
8980 REM
8990 REM
9000 REM
9010 REM
9020 REM
9030 REM
9040 REM
9050 REM
9060 REM
9070 REM
9080 REM
9090 REM
9100 REM
9110 REM
9120 REM
9130 REM
9140 REM
9150 REM
9160 REM
9170 REM
9180 REM
9190 REM
9200 REM
9210 REM
9220 REM
9230 REM
9240 REM
9250 REM
9260 REM
9270 REM
9280 REM
9290 REM
9300 REM
9310 REM
9320 REM
9330 REM
9340 REM
9350 REM
9360 REM
9370 REM
9380 REM
9390 REM
9400 REM
9410 REM
9420 REM
9430 REM
9440 REM
9450 REM
9460 REM
9470 REM
9480 REM
9490 REM
9500 REM
9510 REM
9520 REM
9530 REM
9540 REM
9550 REM
9560 REM
9570 REM
9580 REM
9590 REM
9600 REM
9610 REM
9620 REM
9630 REM
9640 REM
9650 REM
9660 REM
9670 REM
9680 REM
9690 REM
9700 REM
9710 REM
9720 REM
9730 REM
9740 REM
9750 REM
9760 REM
9770 REM
9780 REM
9790 REM
9800 REM
9810 REM
9820 REM
9830 REM
9840 REM
9850 REM
9860 REM
9870 REM
9880 REM
9890 REM
9900 REM
9910 REM
9920 REM
9930 REM
9940 REM
9950 REM
9960 REM
9970 REM
9980 REM
9990 REM

```

HOUGHTS

AND

CROSSES



THIS IS YOUR PIECE



THIS IS THE EX-01 PIECE

**** PRESS NEWLINE TO START ****

(C) S.P.STARTFORD, 28/11/82

THE INTRODUCTION PAGE.

EX-01

00000

HUMAN

00000

DO YOU
WISH TO
GO FIRST?
Y OR N.


**** THE BOARD ****

PEEPER

Someone's watching you! A paranoid little program, this.



FIVE CARD TRICK

You've got to hand it to us — another great game for the ZX81!



This program is a version of the card game, pontoon, for you to type in on your ZX81.

Including all the main features of pontoon, the program starts you off with £50 to gamble and the game will continue until you run out of cash.

TWISTING THE NIGHT AWAY

As in the card game, you will be dealt a single card on which you must decide how much you would like to place a bet. Your second card will then be dealt to you. Should you be dealt two aces, the program automatically assumes the first ace is worth one and asks you whether you

Abstract

PROFIT PROPHET



The object of this program is to help a company or financial institution measure their profitability.

On RUNNING the program you will be greeted with a menu which will provide a route through to the particular ratio you require. Here follows a list of the facilities available with this program:

- a) Return on investment.
- b) Net profit percentage.
- c) Asset turnover.
- d) Gross profit percentage.

- e) Selling expenses/sales.
- f) Administration expenses/sales.
- g) Establishment expenses/sales.
- h) Financial expenses/sales.
- i) Sales and distribution expenses/sales.
- j) Research and development expenses/sales.
- k) Fixed asset turnover.
- l) Current asset turnover.
- m) Working capital turnover.
- n) Debtor's turnover.
- o) Rate of stock turnover.
- p) Current asset ratio.
- q) Acid test.

A serious program to help you get to grips with your business.

- r) Average period of credit given.
- s) Average period of credit received.
- t) Earnings per share.
- u) P.E. ratio.
- v) Dividend per share.
- w) Dividend yield.
- x) Dividend cover.
- y) Capital employed.

As you can see from the long list of options, the program certainly tries to fill in the gaps of your business knowledge.

In fact, the author has added a number of program notes which will be revealed should you choose certain options. These are not part of the operation of the program but illustrate certain patterns which developed over a period of trading months so were added for the author's convenience. Should these not apply to you they could be left out or have new comments substituted.

```

10 GOSUB 9999
20 GOSUB 99
30 CLS
40 PRINT "XCEL PROFITABILITY H
ENTER "
50 PRINT "-----"
60 PRINT "a) RETURN ON INVESTM
ENT"
70 PRINT "b) NET PROFIT PERCENT
AGE"
80 PRINT "c) ASSET TURNOVER"
90 PRINT "d) GROSS PROFIT PERC
ENTAGE"
100 PRINT "e) SELLING EXPENSES/
SALES"
110 PRINT "f) ADMINISTRATION EX
PENSES/SALES"
120 PRINT "g) ESTABLISHMENT EXP
ENSES/SALES"
130 PRINT "h) FINANCIAL EXPENSE
S/SALES"
140 PRINT "i) SALES+DISTRIB.EXP
/SALES"
150 PRINT "ENTER OPTION LETTER
OR JUST NUL"

160 INPUT J%
170 GOTO (CODE J%+100)
180 CLS
190 PRINT "XCEL PROFITABILITY R
AT 100"
200 PRINT "-----"
210 PRINT "j) RES.+DEVELOPMENT
EXP./SALES"
220 PRINT "k) FIXED ASSET TURNO
VER"
230 PRINT "l) CURRENT ASSET TUR
NOVER"
240 PRINT "m) WORKING CAPITAL T
URNOVER"
250 PRINT "n) DEBTORS TURNOVER"
260 PRINT "o) RATE OF STOCK TUR
NOVER"
270 PRINT "p) CURRENT ASSET RAT
IO"
280 PRINT "q) ACID TEST"
290 PRINT "r) AVERAGE PERIOD OF CRED
IT GIVEN"
300 PRINT "ENTER OPTION LETTER
OR JUST NUL"

```

```

210 INPUT JB
220 IF JB=1 THEN GOTO 230
230 GOTO 10000 JB=1000
240 CLS
250 PRINT "1981 PROFITABILITY R
RTIDE"
260 PRINT "-----"
270 PRINT "3) NO. PERIOD OF CRED
ST REC="
280 PRINT "T) EARNINGS PER SHAR
290 PRINT "U) P.E. RATIO"
300 PRINT "V) DIVIDEND PER SHAR
310 PRINT "W) DIVIDEND YIELD"
320 PRINT "X) DIVIDEND COVER"
330 PRINT "Y) CAPITAL EMPLOYED"
340 PRINT "Z) END"
350 PRINT "AT 20.0"
360 PRINT "ENTER OPTION LETTER
OR JUST N/L"
370 INPUT JB
380 IF JB=1 THEN STOP
390 GOTO 10000 JB=1000
400 STOP
4100 CLS
4200 PRINT "RETURN ON INVESTMENT
4300 PRINT "NET PROFIT="
4400 INPUT A1
4500 PRINT H1
4600 PRINT "CAPITAL EMPLOYED="
4700 INPUT C1
4800 PRINT C1
4900 IF A1="?" OR C1="?" THEN RU
5000 PRINT "RETURN ON INVESTMENT
5100 INT (1000 A1/C1+100)
5200 PRINT
5300 GOTO 7000
5400 CLS
5500 PRINT "NET PROFIT PERCENTS
5600 PRINT "NET PROFIT="
5700 INPUT A1
5800 PRINT H1
5900 PRINT "SALES="
6000 INPUT B1
6100 PRINT B1
6200 IF A1="?" OR B1="?" THEN RU
6300 PRINT "NET PROFIT PERCENTAG
6400 INT (1000 A1/B1+100)
6500 PRINT
6600 PRINT "PRESS N/L TO CONTINU
6700 PRINT
6800 PAUSE 100
6900 GOTO 6000
7000 CLS
7100 PRINT "ASSET TURNOVER:"
7200 PRINT "SALES="
7300 INPUT A1
7400 PRINT H1
7500 PRINT "CAPITAL EMPLOYED="
7600 INPUT C1
7700 PRINT C1
7800 IF A1="?" OR C1="?" THEN RU
7900 PRINT "ASSET TURNOVER=" INT
8000 (1000 A1/C1)
8100 PRINT
8200 GOTO 7000
8300 CLS
8400 PRINT "GROSS PROFIT PERCENT
8500 PRINT "GROSS PROFIT="
8600 INPUT A1
8700 PRINT H1
8800 PRINT "SALES="
8900 INPUT B1
9000 PRINT B1
9100 IF A1="?" OR B1="?" THEN RU

```

```

9200 PRINT "GROSS PROFIT PERCENT
9300 INT (1000 A1/B1+100)
9400 PRINT
9500 PRINT "PRESS N/L TO CONTI
9600 PRINT
9700 PAUSE 100
9800 GOTO 6000
9900 CLS
1000 PRINT "SALES EXPENSES/SALES
1010 PRINT "SALES EXPENSES="
1020 INPUT A1
1030 PRINT H1
1040 PRINT "SALES="
1050 INPUT B1
1060 PRINT B1
1070 IF A1="?" OR B1="?" THEN RU
1080 PRINT "SALES EXPENSES/SALES
1090 INT (1000 A1/B1+100)
1100 PRINT
1110 GOTO 7000
1120 CLS
1130 PRINT "ADM. EXPENSES/SALES="
1140 PRINT "ADM. EXPENSES="
1150 INPUT A1
1160 PRINT H1
1170 PRINT "SALES="
1180 INPUT B1
1190 PRINT B1
1200 IF A1="?" OR B1="?" THEN RU
1210 PRINT "ADM. EXPENSES/SALES="
1220 INT (1000 A1/B1+100)
1230 PRINT
1240 GOTO 7000
1250 CLS
1260 PRINT "EST. EXPENSES/SALES="
1270 PRINT "EST. EXPENSES="
1280 INPUT A1
1290 PRINT H1
1300 PRINT "SALES="
1310 INPUT B1
1320 PRINT B1
1330 IF A1="?" OR B1="?" THEN RU
1340 PRINT "EST. EXPENSES/SALES="
1350 INT (1000 A1/B1+100)
1360 PRINT
1370 GOTO 7000
1380 CLS
1390 PRINT "FINANCIAL EXP./SALES
1400 PRINT "FINANCIAL EXPENSES="
1410 INPUT A1
1420 PRINT H1
1430 PRINT "SALES="
1440 INPUT B1
1450 PRINT B1
1460 IF A1="?" OR B1="?" THEN RU
1470 PRINT "FINANCIAL EXP./SALES
1480 INT (1000 A1/B1+100)
1490 PRINT
1500 GOTO 7000
1510 CLS
1520 PRINT "SALES-DISTRIB. EXP./S
1530 PRINT "SALES-DISTRIB. EXP.
1540 INPUT A1
1550 PRINT H1
1560 PRINT "SALES="
1570 INPUT B1
1580 PRINT B1
1590 IF A1="?" OR B1="?" THEN RU
1600 PRINT "SALES-DISTR. EXP./SA
1610 INT (1000 A1/B1+100)
1620 PRINT
1630 GOTO 7000
1640 CLS
1650 PRINT "RESEARCH+DEV. EXP./SA
1660 PRINT "RESEARCH+DEV. EXP.="

```



```

001100 PRINT "*****"
001110 *****
001120 PRINT "IT 18.5"
001130 PRINT "TO CONTINUE"
001140 INPUT " "
001150

```

STOCK CONTROL

Keep efficient control of your stock with this splendid program.



This program for the 16K 2081 allows the handling of stock control files.

Each file set up with deal with up to 100 stock items and will tell you which items require reordering. The program itself is menu driven and offers nine main options: you may enter or delete stock; enter the amount re-ordered; delete files or enter new items; save the file; or print

out a list of all stock items and suppliers via the printer.

The subroutine for each option appears starting at line number 1000 multiplied by the option number. These are called in line 390 by GOTO $1000 * I$, where I is the option chosen.

The option to save a file under a file name allows the facility to run several files,

where each file contains stock of a certain kind. For example, one file could deal with food stock, another with sundries, and a further file could deal with stock for the office. In this way, it would be possible to have as many stock items as you wish on file. A further program could then be written, by the more enthusiastic of you, to index the files and what they contain.

As a last note, the product reference code should be unique to each item and should not be a sub-code of another item. For example, if you enter a code, food, and there already exists a code of this name, the computer will return the first found. However, the entry of the new items routine will tell you if a code already exists, so this should not cause any problems.

SPECTRUM CONVERSION

The program is fairly straightforward and the listing self explanatory, and can be converted to run on the ZX Spectrum with minimal alterations.

First, alter the number in line 100 to 50 for the 16K Spectrum and 250 for the 48K Spectrum. Then, delete all the lines with FAST or SLOW in them. Finally, you may wish to change all the upper case messages to lower case, but this is not absolutely necessary.

```

100 NAME "STOCK"
101 LET Z%="ENTER PRODUCT REFER
102 CODE%=""
103 LET NO=100
104 DIM S%ND,30
105 DIM S%ND,20
106 DIM A%100,4
107 SLOW
108 PRINT TAB 2;"STOCK CONTROL"
109 GOTO 370
370 PRINT "OPTIONS: -"
371 PRINT
372 PRINT TAB 2;"1. ENTER NEW S
373 ITEM"
374 PRINT TAB 2;"2. PRINT ITEM
375 LIST"
376 PRINT TAB 2;"3. DELETE OLD
377 STOCK ITEM"
378 GOTO TAB 2;"4. ENTER RENOU
379 STOCK"

```



```

4138 LET A(1,1)=A(1,1)-R
4139 GOTO 4140
4140 PRINT "ANY FURTHER STOCK"
4141 INPUT Y$
4142 IF DKEY$="" THEN GOTO 4138
4143 IF DKEY$="Y" THEN GOTO 4138
4144 GOTO 4138
4145 CLS
4146 GOTO 300
4147 CLS
4148 PRINT "A"
4149 INPUT B$
4150 IF B$="" THEN GOTO 3000
4151 FOR I=1 TO 40
4152 IF A(I,1) TO LEN B$=B$ THEN
4153 GOTO 4154
4154 NEXT I
4155 GOTO 3070
4156 CLS
4157 GOTO 3070
4158 PRINT "B"
4159 INPUT C$
4160 IF C$="" THEN GOTO 3000
4161 FOR I=1 TO 40
4162 IF A(I,1) TO LEN C$=C$ THEN
4163 GOTO 4164
4164 NEXT I
4165 GOTO 3070
4166 CLS
4167 GOTO 300
4168 CLS
4169 PRINT "C"
4170 INPUT D$
4171 IF D$="" THEN GOTO 3000
4172 FOR I=1 TO 40
4173 IF A(I,1) TO LEN D$=D$ THEN
4174 GOTO 4175
4175 NEXT I
4176 GOTO 3070
4177 CLS
4178 GOTO 300
4179 CLS
4180 PRINT "D"
4181 INPUT E$
4182 IF E$="" THEN GOTO 3000
4183 FOR I=1 TO 40
4184 IF A(I,1) TO LEN E$=E$ THEN
4185 GOTO 4186
4186 NEXT I
4187 GOTO 3070
4188 CLS
4189 GOTO 300
4190 CLS
4191 PRINT "E"
4192 INPUT F$
4193 IF F$="" THEN GOTO 3000
4194 FOR I=1 TO 40
4195 IF A(I,1) TO LEN F$=F$ THEN
4196 GOTO 4197
4197 NEXT I
4198 GOTO 3070
4199 CLS
4200 GOTO 300
4201 CLS
4202 PRINT "F"
4203 INPUT G$
4204 IF G$="" THEN GOTO 3000
4205 FOR I=1 TO 40
4206 IF A(I,1) TO LEN G$=G$ THEN
4207 GOTO 4208
4208 NEXT I
4209 GOTO 3070
4210 CLS
4211 GOTO 300
4212 CLS
4213 PRINT "G"
4214 INPUT H$
4215 IF H$="" THEN GOTO 3000
4216 FOR I=1 TO 40
4217 IF A(I,1) TO LEN H$=H$ THEN
4218 GOTO 4219
4219 NEXT I
4220 GOTO 3070
4221 CLS
4222 GOTO 300
4223 CLS
4224 PRINT "H"
4225 INPUT I$
4226 IF I$="" THEN GOTO 3000
4227 FOR I=1 TO 40
4228 IF A(I,1) TO LEN I$=I$ THEN
4229 GOTO 4230
4230 NEXT I
4231 GOTO 3070
4232 CLS
4233 GOTO 300
4234 CLS
4235 PRINT "I"
4236 INPUT J$
4237 IF J$="" THEN GOTO 3000
4238 FOR I=1 TO 40
4239 IF A(I,1) TO LEN J$=J$ THEN
4240 GOTO 4241
4241 NEXT I
4242 GOTO 3070
4243 CLS
4244 GOTO 300
4245 CLS
4246 PRINT "J"
4247 INPUT K$
4248 IF K$="" THEN GOTO 3000
4249 FOR I=1 TO 40
4250 IF A(I,1) TO LEN K$=K$ THEN
4251 GOTO 4252
4252 NEXT I
4253 GOTO 3070
4254 CLS
4255 GOTO 300
4256 CLS
4257 PRINT "K"
4258 INPUT L$
4259 IF L$="" THEN GOTO 3000
4260 FOR I=1 TO 40
4261 IF A(I,1) TO LEN L$=L$ THEN
4262 GOTO 4263
4263 NEXT I
4264 GOTO 3070
4265 CLS
4266 GOTO 300
4267 CLS
4268 PRINT "L"
4269 INPUT M$
4270 IF M$="" THEN GOTO 3000
4271 FOR I=1 TO 40
4272 IF A(I,1) TO LEN M$=M$ THEN
4273 GOTO 4274
4274 NEXT I
4275 GOTO 3070
4276 CLS
4277 GOTO 300
4278 CLS
4279 PRINT "M"
4280 INPUT N$
4281 IF N$="" THEN GOTO 3000
4282 FOR I=1 TO 40
4283 IF A(I,1) TO LEN N$=N$ THEN
4284 GOTO 4285
4285 NEXT I
4286 GOTO 3070
4287 CLS
4288 GOTO 300
4289 CLS
4290 PRINT "N"
4291 INPUT O$
4292 IF O$="" THEN GOTO 3000
4293 FOR I=1 TO 40
4294 IF A(I,1) TO LEN O$=O$ THEN
4295 GOTO 4296
4296 NEXT I
4297 GOTO 3070
4298 CLS
4299 GOTO 300
4300 CLS
4301 PRINT "O"
4302 INPUT P$
4303 IF P$="" THEN GOTO 3000
4304 FOR I=1 TO 40
4305 IF A(I,1) TO LEN P$=P$ THEN
4306 GOTO 4307
4307 NEXT I
4308 GOTO 3070
4309 CLS
4310 GOTO 300
4311 CLS
4312 PRINT "P"
4313 INPUT Q$
4314 IF Q$="" THEN GOTO 3000
4315 FOR I=1 TO 40
4316 IF A(I,1) TO LEN Q$=Q$ THEN
4317 GOTO 4318
4318 NEXT I
4319 GOTO 3070
4320 CLS
4321 GOTO 300
4322 CLS
4323 PRINT "Q"
4324 INPUT R$
4325 IF R$="" THEN GOTO 3000
4326 FOR I=1 TO 40
4327 IF A(I,1) TO LEN R$=R$ THEN
4328 GOTO 4329
4329 NEXT I
4330 GOTO 3070
4331 CLS
4332 GOTO 300
4333 CLS
4334 PRINT "R"
4335 INPUT S$
4336 IF S$="" THEN GOTO 3000
4337 FOR I=1 TO 40
4338 IF A(I,1) TO LEN S$=S$ THEN
4339 GOTO 4340
4340 NEXT I
4341 GOTO 3070
4342 CLS
4343 GOTO 300
4344 CLS
4345 PRINT "S"
4346 INPUT T$
4347 IF T$="" THEN GOTO 3000
4348 FOR I=1 TO 40
4349 IF A(I,1) TO LEN T$=T$ THEN
4350 GOTO 4351
4351 NEXT I
4352 GOTO 3070
4353 CLS
4354 GOTO 300
4355 CLS
4356 PRINT "T"
4357 INPUT U$
4358 IF U$="" THEN GOTO 3000
4359 FOR I=1 TO 40
4360 IF A(I,1) TO LEN U$=U$ THEN
4361 GOTO 4362
4362 NEXT I
4363 GOTO 3070
4364 CLS
4365 GOTO 300
4366 CLS
4367 PRINT "U"
4368 INPUT V$
4369 IF V$="" THEN GOTO 3000
4370 FOR I=1 TO 40
4371 IF A(I,1) TO LEN V$=V$ THEN
4372 GOTO 4373
4373 NEXT I
4374 GOTO 3070
4375 CLS
4376 GOTO 300
4377 CLS
4378 PRINT "V"
4379 INPUT W$
4380 IF W$="" THEN GOTO 3000
4381 FOR I=1 TO 40
4382 IF A(I,1) TO LEN W$=W$ THEN
4383 GOTO 4384
4384 NEXT I
4385 GOTO 3070
4386 CLS
4387 GOTO 300
4388 CLS
4389 PRINT "W"
4390 INPUT X$
4391 IF X$="" THEN GOTO 3000
4392 FOR I=1 TO 40
4393 IF A(I,1) TO LEN X$=X$ THEN
4394 GOTO 4395
4395 NEXT I
4396 GOTO 3070
4397 CLS
4398 GOTO 300
4399 CLS
4400 PRINT "X"
4401 INPUT Y$
4402 IF Y$="" THEN GOTO 3000
4403 FOR I=1 TO 40
4404 IF A(I,1) TO LEN Y$=Y$ THEN
4405 GOTO 4406
4406 NEXT I
4407 GOTO 3070
4408 CLS
4409 GOTO 300
4410 CLS
4411 PRINT "Y"
4412 INPUT Z$
4413 IF Z$="" THEN GOTO 3000
4414 FOR I=1 TO 40
4415 IF A(I,1) TO LEN Z$=Z$ THEN
4416 GOTO 4417
4417 NEXT I
4418 GOTO 3070
4419 CLS
4420 GOTO 300
4421 CLS
4422 PRINT "Z"
4423 INPUT AA$
4424 IF AA$="" THEN GOTO 3000
4425 FOR I=1 TO 40
4426 IF A(I,1) TO LEN AA$=AA$ THEN
4427 GOTO 4428
4428 NEXT I
4429 GOTO 3070
4430 CLS
4431 GOTO 300
4432 CLS
4433 PRINT "AA"
4434 INPUT AB$
4435 IF AB$="" THEN GOTO 3000
4436 FOR I=1 TO 40
4437 IF A(I,1) TO LEN AB$=AB$ THEN
4438 GOTO 4439
4439 NEXT I
4440 GOTO 3070
4441 CLS
4442 GOTO 300
4443 CLS
4444 PRINT "AB"
4445 INPUT AC$
4446 IF AC$="" THEN GOTO 3000
4447 FOR I=1 TO 40
4448 IF A(I,1) TO LEN AC$=AC$ THEN
4449 GOTO 4450
4450 NEXT I
4451 GOTO 3070
4452 CLS
4453 GOTO 300
4454 CLS
4455 PRINT "AC"
4456 INPUT AD$
4457 IF AD$="" THEN GOTO 3000
4458 FOR I=1 TO 40
4459 IF A(I,1) TO LEN AD$=AD$ THEN
4460 GOTO 4461
4461 NEXT I
4462 GOTO 3070
4463 CLS
4464 GOTO 300
4465 CLS
4466 PRINT "AD"
4467 INPUT AE$
4468 IF AE$="" THEN GOTO 3000
4469 FOR I=1 TO 40
4470 IF A(I,1) TO LEN AE$=AE$ THEN
4471 GOTO 4472
4472 NEXT I
4473 GOTO 3070
4474 CLS
4475 GOTO 300
4476 CLS
4477 PRINT "AE"
4478 INPUT AF$
4479 IF AF$="" THEN GOTO 3000
4480 FOR I=1 TO 40
4481 IF A(I,1) TO LEN AF$=AF$ THEN
4482 GOTO 4483
4483 NEXT I
4484 GOTO 3070
4485 CLS
4486 GOTO 300
4487 CLS
4488 PRINT "AF"
4489 INPUT AG$
4490 IF AG$="" THEN GOTO 3000
4491 FOR I=1 TO 40
4492 IF A(I,1) TO LEN AG$=AG$ THEN
4493 GOTO 4494
4494 NEXT I
4495 GOTO 3070
4496 CLS
4497 GOTO 300
4498 CLS
4499 PRINT "AG"
4500 INPUT AH$
4501 IF AH$="" THEN GOTO 3000
4502 FOR I=1 TO 40
4503 IF A(I,1) TO LEN AH$=AH$ THEN
45
```

```

0000 GOTO 000
0001 KEY
0002 PRINT THE S-STORE STOCK FILE
0003
0004 PRINT "ENTERED FILE NAME."
0005 INPUT #1
0006 IF #1="" THEN GOTO 0000
0007 PRINT #1
0008 PRINT "START TAKE REORDER"
0009 PRINT "ORDER ADVISE TO 5000 FILE."
0010 IF INKEY#="" THEN GOTO 0007
0011
0012 IF INKEY#="" THEN GOTO 0000
0013 PRINT #1
0014 GOTO 0000
0015 PRINT "FILE :#0"
0016 PRINT
0017 GOTO 0000
0018 PRINT
0019 PRINT "STOCK/SUPPLIERS LIST"
0020
0021 FOR I=1 TO 10
0022 IF #11. TO #10" - THE
0023 GOTO 0000
0024 PRINT
0025 PRINT "STOCK REP. CODE."
0026 PRINT #111
0027 PRINT "SUPPLIER."
0028 PRINT #112
0029 PRINT I
0030 PRINT
0031 GOTO 0000
0032 GOTO 0000

```

STUDY CONTINUED

RETURN:

- ```

ENTER NEW STOCK ITEM.
PRINT ITEM DETAILS.
DELETE OLD STOCK ITEM.
ENTER REMOVED STOCK.
ENTER STOCK ON ORDER.
ENTER STOCK RECEIVED.
PRINT ITEMS TO RE-ORDER.
PRINT ALL STOCK FILE.
PRINT ALL STOCK CODES
END.

```

CHERRY OFFICE 01-51

| ENTER SUBJECT REFERENCE CODE | POSTAL ADDRESS | CITY | STATE | ZIP |
|------------------------------|----------------|------|-------|-----|
| 100                          | 100            | 100  | 100   | 100 |

**ENTER AMOUNT IN STOCK**



**Abstract**

ENTER PG-GRADER LEVEL.

| Year | U.S. should take action (%) | U.S. should not take action (%) |
|------|-----------------------------|---------------------------------|
| 1994 | 75                          | 25                              |
| 1996 | 85                          | 15                              |
| 1998 | 75                          | 25                              |
| 2000 | 85                          | 15                              |
| 2002 | 75                          | 25                              |
| 2004 | 85                          | 15                              |

RECEIVED BY:  NO.  DATE:

ENTER SUPPLIER'S DETAILS.  
B. BARTON, LTD.

SEE FURTHER NEW ITEMS p. 66-67

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

**PUBLISHED BY**

DIVISION STOCK : 03  
SUBSTOCK ON ORDER : 00  
QTY - QTY OF LEVEL : 00  
QTY - QTY OF QUANTITY : 00

10. **NAME OF THE PARTY:** \_\_\_\_\_  
 11. **DATE OF THE PARTY:** \_\_\_\_\_

PLEASE DO NOT FOR COPY. ANY OTHER  
COPYING IS PROHIBITED.

## PRODUCTS BEAUTY INC. F-0000100

[illegible]

# AIR RAID

In this program, Stephen has taken trouble to incorporate moving graphics, simple rules and an emphasis on strategy rather than relying on Lady Luck.

The rules of the game are quite simple. You begin the game with 30 bombs for your aircraft and you can release a bomb by pressing the 'O' key. Your aim is to hit all the enemy ships and planes you can within the restriction of how many bombs you have. If a ship reaches the landing stage before you can destroy it you will lose one of your valuable bombs, and you are further restricted by only being able to fire one bomb at a time (if there is a bomb still on-screen, you will not be allowed to fire another).

the sea harbour — you travel across the screen from right to left, and when you disappear on the left-hand side you miraculously appear on the right-hand side again. The white speck in the centre of your aircraft indicates that you have a bomb on the board.

Ships will appear on the left-hand side of the screen and slowly move toward the landing stage. As you move across the screen in your aircraft, you must decide when to release your bomb in order to destroy the ship. If you manage to score a direct hit on the ship, you will be rewarded with the ship being swallowed up by the mysterious deep. However, if you miss the ship, the bomb will explode

**Run for cover — and have a smashing time.**

tactic you could employ is the 'shot in the dark' option in which you release a bomb before a ship has yet come onto the screen. But remember, you have only 30 bombs, so it's best to make them count.

When you have used up your 30 bombs, the display will scroll upwards so that the sea ends up at the top of the screen. The score you have achieved during the game is then displayed, followed by the top highest scores of the day. The sea will then begin rolling again and the CGM will wait for you to press 'O' signifying that you would like another game.

To help you decipher the program listing, perhaps the



## PLAIN SAILING

When you first RUN the program, the Score Advance Table will appear on the screen. From this you will be able to see that ships are more valuable (in terms of how many points are scored) to destroy than planes. The aircraft are not easy to hit — but there is no penalty for allowing a plane to escape your fire.

After pressing the 'S' key, the score table is erased and you should get your finger over the 'O' key — this is the only control key used in the game. Your ammunition will appear as a series of 'grated railings' at the top of the screen. The sea will appear at the bottom of the screen with a landing stage in the bottom right-hand corner. You are positioned in a aircraft, aimed just below your ammunition dump at the top of the screen.

Your aircraft slowly 'circles'

harmlessly in the sea and the ship will dock at the landing stage, its troops will disembark unopposed and steal one of your bombs.

Periodically, an enemy plane will fly across the screen from left to right, moving twice as quickly as their naval colleagues. They are not trying to reach the landing stage, their object is to get between you and the ships, thus blocking your bomb. There will only ever be one ship or one plane visible on the screen at any time, but a plane and a ship may be seen together.

## A SHOT IN THE DARK

One of the problems you will encounter is judging when to release your bombs so that they will destroy the ships. One

following would be useful. The ships and aircraft are held in string arrays, A3, B5 and C3. Cruisers, represented by A5, are generated as follows:

A3(1) — "Space, Graphic 8,  
Graphic 4, Space"  
A3(2) — "Space, Graphic 8,  
Graphic 5, Space"  
A3(3) — "Graphic 8, Inverse

Space, Inverse Space, Graphic 8"

B5 is used to represent the aircraft and is comprised of the following:

B5 — "Graphic 7, Graphic 7,  
Graphic 6, Graphic 5"

Battleships are held in C3 and are made up as below:

C3(1) — "3 Spaces, Graphic 3,



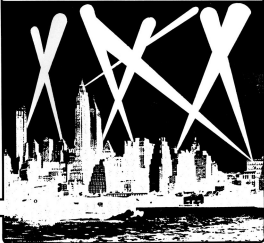




```

0004 PRINT TAB 8;"-- ENEMY AIRBOR
0005 GOTO 0010
0006 PRINT
0007 FOR M=1 TO 4
0008 PRINT "0800:"
0009 NEXT M
0010 PRINT AT 8,10;"100,000 OR 0
0011 GOTO 0015
0012 PRINT AT 8,9;"-- BATTLESHIP
0013 GOTO 0015
0014 PRINT AT 12,8:
0015 FOR M=1 TO 5
0016 PRINT "0900:"
0017 NEXT M
0018 PRINT AT 12,10;"200,400 OR
0019 GOTO 0025
0020 PRINT AT 14,7;"-- TROOPS ON
0021 AIRBOR
0022 PRINT AT 17,6;"PRESS ""5""
0023 TO STOP
0024 IF INKEYS=""5" THEN GOTO 01
0025 CLS
0026 RETURN
0027 PRINT AT 8,8;"OUT OF RANGE
0028 TCON
0029 FOR L=1 TO 50
0030 NEXT L
0031 FOR L=1 TO 50
0032 GOTO 0005
0033 GOTO 0005
0034 PRINT AT 3,8;"YOUR SCORE WA
0035 0
0036 LET YU=0
0037 FOR M=1 TO 5
0038 IF 50/YU=0 THEN GOTO 0005
0039 NEXT M
0040 PRINT AT 5,9;"TODAY""S SIX-
0041 OF-THE-BEST
0042 FOR M=1 TO 5
0043 PRINT AT M+5,5,M:
0044 NEXT M
0045 PRINT AT 14,9;"PRESS ""5""
0046 TO PLAY AIR RAID
0047 GOTO 0005
0048 GOTO 0005
0049 IF INKEYS=""5" THEN GOTO 01
0050 PRINT
0051 CLS
0052 GOTO 0040
0053 FOR L=5 TO M+1 STEP -1
0054 LET YU=YU-L-1
0055 NEXT L
0056 LET YU=50
0057 PRINT AT M+5,15;"WELL DON
0058 0
0059 GOTO 0005

```



# CODING GRAPHICS

Increase your character size with this great utility.

The purpose of the program is to enable screen displays of large characters or user-defined graphics to be created quickly and easily by calling the character or graphic from a library of symbols stored in an array. On program termination, the screen display is saved in high memory, and is then available for inclusion in a new program.

The program is entertaining to use, and some interesting effects can be achieved by mixing both normal and inverse characters on the screen. The symbol library can hold 54 characters, each designed on a three by five matrix, and as an aid, I enclose a list showing the numeric data required to create 54 possible characters. The program may be used with an incomplete library, any new characters being added as and when required, with the option to re-define any previously created character.

## HOW THE PROGRAM WORKS

Lines 20 to 240 are concerned with creating a graphic character, and storing the associated data in an array. The array subscript C is calculated by the subroutine at line 640, after which a check is made at line 100 to see if the character has already been defined. When character definition is completed press Newline to enter the PLOT routine, which works as follows.

Lines 260 and 270 set the initial printing position for the first character to be plotted. Line 290 requests a character, the subroutine at line 640 converting C8 onto the appropriate location C in the array. The individual bytes



stored in this string location are then POKE'd into the first 15 locations of the line 1 REM statement. This operation is carried out by lines 320 to 390.

Line 410 copies the screen display to above RAMtop. Line 420 puts the chosen character at the initial print position. Lines 430 to 470 scan the keyboard for an input, with any X-Y moves incrementing or decrementing the values held in

addresses 16542 and 16544.

Line 500 copies the screen display back from above RAMtop, a jump then being made to the 420 where the chosen character is re-printed. Line 470 permits an escape from this loop, enabling the chosen character to be incorporated into the screen display when line 410 is next executed.

When character plotting is completed, pressing Newline

enters the insert routine, executed by line 500. Line 570 stores the screen display above RAMtop. The final option is to finish the program by typing T, or Newline to re-enter the plotting routine.

## CREATING THE PROGRAM

Reserve some memory for the machine code routines by typing the following direct commands:

```
POKE 16389,0
POKE 16389,125
NEW
```

This sets RAMtop to 32000. Type in the Hex loader program figure 1, ensuring that line 1 contains at least 128 characters. Enter the Hex code as shown in the left-hand column of figure 2, terminating each line of code with Newline. If a mistake is made in typing, pressing Newline will select 'edit' mode indicated by an inverse 'I' being displayed. Now type the correct code and continue, or re-edit as required. Terminate data entry by typing Z. Delete all lines except line 1, then press Clear.

As a direct command, type DIM A\$(54,5,3) to create an array of 54 characters. Note that this direct command avoids the use of a line number in the program, and so careless use of GOTO cannot accidentally reDIMension the array and destroy previously entered data. RUN will also destroy data, and so the program should always be restarted using GOTO 1. Add the BASIC lines such that the program is now as shown in figure 3. Type GOTO 1 and use figure 4 to create the characters

required, entering a number followed by Newline for each of the 15 entries per character. The character is built up and displayed as each entry is made. When data entry is completed, opt out of this routine by pressing Newline. Step through the options by successful presses of Newline, typing 'Y' in response to 'EXIT PROGRAM?'. To save the program on tape, start the cassette recording, then type GOTO 620.

## USING THE PROGRAM

Ensure the RAMtop is set to 32000. The program will start automatically when loaded, but if a re-start is required type GOTO 1. In response to 'CREATE WHICH CHARACTER?' press Newline. The next reply will produce the chosen character in the lower left-hand corner of the screen. Steer the character into place and fix using the keys as prompted.

The next character selected will be initially superimposed on the previous one, and may be manipulated as before. You can opt out of this routine by pressing Newline. For inverse graphics press 'Y' in response to the prompt. A 'Y' response to the next prompt will terminate the program. A screen dump (Fig. 5) shows typical characters. The display has now been saved in high memory. To call this display into a new program press New, then enter the BASIC shown in Fig. 6. Run this program, and when the report code 070 appears, each program line may be deleted. The graphics may now be displayed, either by a direct command PRINT 25 or as a

program line, eg 10 PRINT 25 followed by GOTO 10.

## A LOAD OF CODE

The machine code routines have starting addresses as follows:

```
16629 C.UP
16641 PRINT
16661 C.DOWN
16604 INVRT
```

The routines C.UP and C.DOWN use the LDIR instruction to block-move 727 bytes of code from the address of the display file to address 32000 and vice versa. The Print routine works as follows. The B and A registers hold the X and Y printing offset with respect to screen 0,0. Steps 19 and 30 increment the display file address by an amount equal to the value in register B. Steps 33 and 35 then increment the display file address by an amount equal to 33 times the value in register A.

Having established the printing position, steps 31 to 34 print the first three bytes of the enlarged character. Steps 35 to 37 then increment the display file address by 30 point to the corresponding printing position on the next line, and the next three bytes are then printed here. This process is repeated five times to print the complete character, at which time the C register, previously loaded with five at step 27, has been incremented to zero and a return to BASIC is made.

The INVRT routine finds each printing character in the display file, adds 128 to the current value, then overwrites the old value.

```

1 REM1.....2....
.....3.....4.....5.....
.....6.....7.....8.....
.....9.....A.....B.....
10 LET S=16614
20 INPUT RS
20 IF RS="" OR INT LEN RS/2 <
 LEN RS/2 THEN GOTO 20
30 PRINT RS;S;
40 INPUT Z
50 IF INT LEN Z/2 < LEN Z/2
 THEN GOTO 20
60 IF Z="" THEN GOTO 100
70 IF Z="" THEN GOTO 100
80 PRINT AT 21,0;CHR
90 GOTO 20
100 SCROLL
110 POKE X,16+CODE RS+CODE RS/2
120 LET X=X+1
130 IF RS="" THEN GOTO 1
140 IF RS="" THEN GOTO 110
150 IF RS="" THEN STOP
160 LET RS=""
170 GOTO 40

```

Fig. 1. The Hex loader program.

Fig. 1. The flow cycle. — liquid inside glass cell inside from the left-hand column, and flow across flow-line.

Fig. 2. A series of drawings showing typical characters designed using this program.

**Fig. 4. The number, percent and the BASIC percentage**



# SPIRALLING INTO GRAPHICS

The author has developed a fascinating spirograph program which even allows 'impossible' patterns to be generated.

**S**pirograph patterns are formed by both the interior and exterior epicycloid curves.

There are two basic equations for these curves:

**INTERIOR EPICYCLOID:**

$$X = (A-B) \cos I + H \cos$$

$$(A-B)/B)$$

$$Y = (A-B) \sin I - H \sin (A-B)$$

$$/B)$$

**EXTERIOR EPICYCLOID:**

$$X = (A+B) \cos I - H \cos$$

$$(A+B)/B)$$

$$Y = (A+B) \sin I - H \sin$$

$$(A+B)/B)$$

where A is the radius of the large circle (circle A), B is the radius of the small circle (circle B) and H is a point on the circumference of the small circle.

## THE EPICYCLOID

This is the curve traced out by the point H on the circumference of the small circle, which rolls on the inside or outside of the large circle. As with the commercial toy, 'Spirograph', the point H can be moved from the circumference to any point along the radius to the centre, where it will just draw a circle.

When the point H is nearer the circumference of the small

circle, the patterns tend to be geometric in shape. When point H is moved nearer to the centre of circle B, the curve tends to be more floral, or organic in shape. The general program for the interior epicycloid is:

```

1 INPUT A
2 INPUT B
3 INPUT H
10 FOR I = 0 to 2 *PI STEP
PI/10
20 LET X=(A-B)*COS I+H
* COS (A-B)/B)
30 LET Y=(A-B)*SIN I-H*SIN
(A-B)/B)
40 PLOT 30+X,22+Y
50 NEXT I

```



Some examples of the sort of patterns that can be generated.



## DRAWING PATTERNS

The basic shapes of the hexagon, pentagon, square, triangle and ellipse can easily be entered into this program. Anything higher than a hexagon tends to become indistinct. To draw other patterns, the value for I has to be changed. It varies, on the Z801, from two to 36 times PI.

There are a few ways of increasing the variety of patterns, but with low resolution graphics, the number is restricted. The three best ways of increasing the variety of patterns is:

1. 'Moving to the centre', that is, moving the point H from the circumference of circle B to near its centre. Try this, with A set equal to 20 and B to 10.

3. FOR H=2 TO 10 Step 4:60  
NEXT H

2. 'Moving to the side', that is, rotation of the pattern, which can be done by moving an angle, which must be added to the general program. Try 2, with A equal 20, B equal 10, H equal to 0, and with the loop FOR I to 2 \*PI STEP PI/10. You'll need to add the following:

```

4. FOR I=0 TO 3*PI/2
20 LET X=(A-B)*COS
I+H*COS(A-B)/B+I)
30 LET Y=(A-B)*SIN
I+H*SIN(A-B)/B+I)
60 NEXT I

```

3. 'Moving to the centre and the

side', that is, movement of point H and rotation. The program is as in two, with the addition of:

```
3 LET H=0
6 LET H=H + 2
```

There is also the 'extended internal pattern' which is achieved by adding the following to the general program:

```
3 LET F=10
8 FOR H=3 TO 8 STEP 2
9 LET F=F*40: PLOT F + X,
22 + Y
60 NEXT H
```

## RATIOS

Although the low resolution graphics tend to be limiting, the computer offers experimentation in other directions, which are not available on a real Spirograph. Besides the obvious one of a greater variety of ratios, there is the possibility of putting the smaller number in first, and the bigger number in

second, as though the larger circle is rolling inside the smaller one.

The following ratios will fit easily into the general program:

```
A 4 6 6 7 8 9
B 13 13 13 13 13 13
H 30 10 10 10 10 10
```

And for these use FOR I=0 TO 267H

```
A 4 7 8
B 9 9 9
C 8 8 8
```

For these use FOR I=0 TO 187H

```
A 4 6 8
B 11 11 11
H 9 9 9
```

And for these we suggest FOR I=0 TO 227H

Some of these have an almost mandala-like quality.

The following can be used in the exterior equations:

```
A 4 5 6
B 7 7 7
H 6 6 6
```

And use FOR I=0 TO 147H

```
A 4 5
B 9 9
H 6 7
```

FOR I=0 TO 187H

## EXPERIMENTS

Another direction for experimentation is based on the moving point H, program one. Put the H loop into the I loop. This will draw the points of H first (FOR H=0 TO 6).

And finally, a way of showing the smaller circle moving inside the larger circle, by using the rotation program, is as follows. Put the I loop inside the J loop. This will draw just six circles. Use FOR I=0 TO 27H STEP P/10 and FOR J=0 TO 27H STEP P/10.

## SPECTRUM

These super value for money cassettes are only available via MAIL ORDER from ORWIN SOFTWARE

## ZX-81

### FOR THE SPECTRUM

(will also run on ORW Spectrum)



**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

**EDUCATIONAL COMPENDIUM FOR THE SPECTRUM**  
All these programs on one cassette.

# SHOOTER

Simple but fun, that's how we view this game.



**H**ere is a program for the 16K Z801 which combines machine code with BASIC to form a simple, but fun to play, game.

In this listing you will first need to enter the machine code using a machine code loader program such as the one

included in this article. You will need 16K of memory because the machine code looks for 118, the Newline code, which means that unless RAMtop is more than 3.5K the machine code will not allow you to fire at the target.

To enter the machine code,

first enter the following program:

```
1 REM
2 POKE 16810,0
3 FOR N=0 TO 11
4 INPUT I
5 POKE (16814+N),I
6 NEXT I
```

Now, RUN the program and input the following machine code (the commas separating the numbers represent you pressing Newline):

42,14,64,126,204,118,200,  
54,22,35,24,247.

Now, remove all the lines except line 0, and type in the main body of the program shown below.

## ON TARGET

Once you have typed in the whole program, type RUN and you are ready to start the game. On the right of the screen, the target will appear and on the left a column of arrows will be seen. When the arrow is in line with the target, press the 'F' key to fire.

You get ten goes to hit the target, after which you will be given your score.

```
0 REM ***** SHOOTER *****
1 LET S=0
2 FOR N=0 TO 10
3 LET L=0
4 LET A=INT (16814+N*5)
5 PRINT AT A,23;" "
6 FOR M=0 TO 25
7 NEXT M
8 FOR M=1 TO 25
9 PRINT AT M,2;" "
10 IF INKEY$="F" THEN LET L=L+5
11 IF L=10 THEN GOTO 25
12 NEXT M
13 IF M=0 THEN LET S=S+5
14 NEXT M
15 PRINT "SCORE=";S;" OUT OF 5"
16 PRINT AT 25,2;"PLAY AGAIN"
```

```
17 GOTO 1
18 IF INKEY$="Q" THEN GOTO 4
19 IF INKEY$="P" THEN GOTO 100
```

```
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

A screen illustration from the program, Shooter.



# ORIC & ATMOS 48K Cassettes



**Super 3D invaders with laser gun and 3D colour** (rotation to their needs). 8 rapid colour transfer screens of increasing difficulty with keyboard controlled 3D laser movement and fire sequence. (Screen required) **£5.95**

## Drills (Smart Dray)

Hours of family fun. One player controls the target, another gets with the laser. (Screen required) **£5.95** (Also in French)

## Mystery Tower

Super music arrangements, 47 different rooms, 3D and colour make this adventure game most enjoyable. How efficiently get lost in target where you're going. **£5.95** (also in French)



## Flight Simulator

With engine, radar ground, vertical flight path indicators, artificial horizon and altimeter with continuous flight indicator on engine conditions, stall speed etc., full colour hi-res graphics, 3D and colour. **£5.95**



## Space Crystal

Space trading adventure game. Fascinating 3D space travel on your voyage mission of the future crystal, black holes, nebulae storms, space pirates, etc. Full colour and sound. **£5.95**. Also available for Dragon 32, Spectrum and Commodore 64



CLARK DATA P.O. BOX 51, Swadlow, Wills. Tel. 07763 45061.

Please supply:

.....

.....

.....

Name

Address



# Little Brothers should be seen but not heard.



REGULAR CONDENSED OR EXTENDED LINES.



EXTENDED SPACES BOLD FONT.



BEST COPY AVAILABLE AVAILABLE.

A maxim which eloquently describes the Brother HR-5.

Less than a foot across, it's nonetheless loaded with features.

But there's one thing the HR-5 won't give you. Easiness.

For the annoying 'clickety click' many printers produce is mercifully absent from the HR-5.

Quietly efficient, it delivers high definition dot matrix text over 80 columns at 30 c.p.s.

The HR-5 also has something of an artistic bent.

Being capable of producing uni-directional graph and chart images together with bi-directional text.

It will also hone down characters into a condensed face, or extend them for added emphasis.

Incorporating either a Centronics parallel or RS-232C interface, the HR-5 is compatible with

most home computers and popular software.

Perfectly portable, the battery or mains operated HR-5 weighs less than 4lbs, and has a starting price of only £179.95 (inc.VAT).

Which is really something to shout about.

PLEASE SEND NO MORE DETAILS OF THIS PLEASING BROTHER HR-5 PRINTER.

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

