

Personal SOFTWARE

**Get more
from your
Spectrum**



- Great games to test your reflexes
- Learn morse code
- Go for different graphics
- Play cards, win wars and destroy aliens
- Useful utilities to run

ANIROG SOFTWARE

COMPUTER GAMES
OF TOMORROW
AVAILABLE

NOW!

AS SEEN ON
T.V.!



HEXPERT



MOON BUGGY



SKRAMBLE



3D TIME TREK

KONG K.B./J.S. £7.95

3 KONGS WITH WILLIAMS PATENTED GAMES'S

SKRAMBLE J.S. £7.95

8 SECTIONS TO TEST YOUR SKILL

HEXPERT J.S. £7.95

TRIALS LEFT TO BE HEXPERTS ON THIS 3D HEXAGONAL PLATFORM

MOON BUGGY J.S. £7.95

DISCOVER YOUR PATH, GRANT OVER GRANT POT HILLS AS

YOU DEFEND THE MOON BUGGY FROM ALIEN ATTACK

FROG RUN K.B./J.S. £5.95

JOYFUL ARCADE GAME, KLB FOR THE WHOLE FAMILY

3D TIME TREK K.B./J.S. £5.95

OFFY TALKING 3D GRAPHICS, 3D TIME TREK GAME

DUNGEONS K.B. £6.95

ENTER THE REALM OF DUNGEONS IN THIS ROLE PLAYING GAME

DARK DUNGEONS K.B. £6.95

2ND & 3RD LEVELS OF POWER, 3D GRAPHICS FOR THE FANTASY

WORLD

COMMODORE 64

NEW

GALAXY

AVOID CAPTURE BY THE ALIEN

MOTHER SHIPS TRACTOR BEAM AS

THE FIGHTING CRUISERS FOR

100 SCREENS WITH A 2-PLAYER

OPTION K.B./J.S. £7.95

NEW APPROVED GAMES FOR
THE STACK LIGHT RIFLE

INDIAN ATTACK £5.95

COSMIC COMMANDO £5.95

VENGEANCE OF ZENO £5.95

BANK, CREDIT CARD SALES, MONLEY (STON) MON
PAYMENT BY CHEQUE, P.O., ACCESS/VISA
8 HIGH STREET, MONLEY, SURREY

Overseas 50p post & packaging

TRADE ENQUIRIES WELCOME

28 West Hill, Dartford, Kent, ID322J92673/8

Get more from your Spectrum



Volume 2 No 4 Spring 1984

Editor: Wendy J Palmer
Sales Executive:
David Fowler
Advertisement Manager:
Barry Bingham
Advertisement Copy:
Christine
Art Director:
Alan McDermott
Managing Editor:
Ron Harris
Chief Executive: T J Connell
Origination and design by:
HBF Design & Print

Personal Software is a quarterly magazine appearing on the third Friday in May, August, November and February.

Contributors to: **Amiga Press** (Sales & Distribution), Ltd., 12, 18 East Street, Chichester, W Sussex, PO19 1AB, Tel: 01243 8222.
Contact by: **Adamsky Press** & Sons Ltd, 100, North, Manchester, Lanc.

The contents of this publication (including all articles, diagrams, plans, drawings and programs) and all copyright and other intellectual property rights therein belong to **Amiga Press** (Sales & Distribution), Ltd. All rights reserved by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to **Amiga Press** (Sales & Distribution), Ltd. and any reproduction without the prior written consent of the Company, © 1984 **Amiga Press** (Sales & Distribution), Ltd.

Subscription Rates: £10.00 for four issues including postage. Payment and other correspondence to: **Personal Software**, Advertising Dept., 101 New St, London, EC2A 3DF. Tel: 01-437 0626.

CONTENTS

Editorial & Advertisement Office
No 1 Golden Square, London W1R 3AB
Telephone: 01-437 0626 Telex: 8811896

Features	4	Alien	89
Battleships	5	Stop the alien force or you could find yourself in trouble.	
Command the fleet and sink the enemy's ships.		Leprechaun's Gold	92
Tourist Trap	18	Are you lucky enough to find the gold?	
Ever had one of those disastrous holidays fraught with problems? See how you would cope.		Circuit Sketch	95
Pinet	19	Great definition when drawing circuit diagrams.	
An educational game for the Spectrum.		Isis	98
Play Your Cards Right	22	A batty little game to get your Spectrum going.	
See if you can deal yourself a good hand.		ED Houghts and Crosses	103
Demolition	26	Throw away your pencil and paper and play on your Spectrum.	
Get rid of your destructive tendencies without having to pay for the damage.		Air Raid	105
Drawing on Experience	30	This game will have you shooting down the enemy planes and missiles.	
A tutorial on the subject of on-screen drawing including a bar chart program and a rather sophisticated sketching utility.		Demolition	72
Earth Attack	31	This game can quickly drain your energies.	
Can you beat off the attack and save the earth?		Machine Code Colour Graphics	74
Toolbox Trio	38	Machine Code isn't really difficult — see our colour graphics.	
Three useful routines for your Spectrum.		Getting Physical	79
Spectrum Breakout	39	Make money by selling lemonade.	
Demolish the wall and be as aggressive as you like.		POKEing the Spectrum Display	80
Alternative Characters	40	What a fine display this utility makes.	
Feed up with the characters on the Spectrum? Re-define them with this utility.		Asteroids Ahead	83
Statistical Analyst	41	Steer clear of the asteroids and keep your spaceship flying.	
Get to grips with numbers and statistics.		Learner Driver	88
Morse Trainer	48	Will you pass the test?	
Talk to your Spectrum in morse code.		Defending Your Spectrum	88
		Let this game knock down your defences.	
		Spectrum	90
		An incredible disassembler for the 48K Spectrum.	

Credits: Our grateful thanks to the following individuals for their assistance in the production of this issue: Mark, Gary, Bill, Bill, Nigel, Tom, William, Adrian, Ivan, Peter and others about too numerous to mention.

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

PREFACE

All right then, hands up everyone who received a Spectrum from Santa Claus this Christmas. And hands up all of you who have beaten the games that you bought to play on your new toy.

It is true to say that with micros, rage spread, forever addicted. And you can't wait to get home with the new game you've just bought to see if you can defend the earth, destroy the aliens and sink the enemy's battleships — and all before tea! The Spectrum is so popular with the home micro user that there is no lack of commercial software available for you to test

your wits and reflexes. Sadly, though, few of us have a bottomless pocket and the standard of software around is not universally high.

It is also probably true to say that although a large number of Spectrum owners start their computing careers wanting to play games, they soon learn that there is more to micros than just splatting aliens. They want to find out more about how their Spectrum works, and they start tentatively writing their own programs (games included).

In this issue of *Personal Software* we have drawn together some of the best programs for the Spectrum that have been published in our sister magazine, *ZX Computing*.

The following pages are packed with games for you to type in — from the more 'old-fashioned' game of noughts and crosses to the current trend of Aliens and Earth Attack. For those who are not particularly interested in games, there are some useful utilities and programs such as Spectramon and Alternative Characters. Certainly there should be something for everyone in this issue!

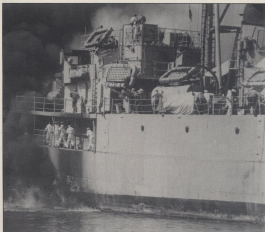
CHECKING UP

We have tried to ensure that the programs in this issue are error-free, and that they are legible, so if you find that you are having problems with any of them, please check them again thoroughly, or get someone else to check them for you. If as a last resort you can still not get the program running, then write to *Personal Software* with a SASE. Please note however that we are unable to answer telephone enquiries.



BATTLESHIPS

Command your own fleet and sink the enemy's ships.



This program has been written to allow users to challenge their Spectrum computer to the age-old classroom game of Battleships. When you have the game **RUNNING**, you will be asked to **INPUT** the direction of your ships and the starting square. The Spectrum will automatically set your ships on the grid. Once, a battleship (four squares), two cruisers (three squares), two destroyers (two squares) have been set up, the Spectrum decides who should begin the game.

The Spectrum's moves are calculated in lines 1000 to 1080; a random square is chosen to fire at unless the Spectrum has just scored a hit on your fleet. If it has just achieved damage to one of your ships, it will continue to fire at the surrounding squares until it has sunk the vessel.

IN THE NAVY

Once the Spectrum has selected a square to fire at, it is up to the user to respond with either a 'M' for a miss or an 'H' for a direct hit. There is no routine in the

program to counteract the user lying to the computer, but there is no reason why a simple routine could not be written to stop any 'innocent' cheating.

When it is the user's turn, simply **INPUT** the co-ordinates, for example, 'F7' for the grid reference F7, and the result will be displayed on the screen as a miss or as a hit. It might be an idea to make a mental note as to which squares you have already fired at, as the computer will not tell you if you have already had a go at a certain grid reference.

```

0000 00 SUB 8000 REM ** RULES **
0001 00 SUB 8000 REM ** INITIAL
THE MONSTER **
0002 00 SUB 8000 REM ** 2
ST 10000 **
0003 00 SUB 8000 REM ** SET UP
0004 00 SUB 8000 REM **
0005 00 SUB 8000 REM ** HIDE ON
0006 **
0007 LET I=INT (RND*2): IF I=1 Y
0008 GO TO 100
0009 GO TO 100: REM ** PLAYER GO
0010 **
0011 00 TO 1000: REM ** SPECTRUM
0012 **
0013 PRINT AT 14,14: PAPER 1: IN
YOUR CO*
0014 YOUR D4: IF LBN D4=12 THEN
GO TO 718
0015 LET D4=D4+1: LET C=AL, D4(
2)
0016 LET D=CODE D4-D4
0017 IF D=0 OR D=1 OR D=2 OR D=3 OR
D=4 THEN GO TO 718
0018 LET U=0: LET A=D+1
0019 LET J=IN, D1
0020 IF MIN, U, D1+3 THEN LET HIT
0021
0022 PRINT INK 2: AT C,D: CHR$ J:
IF
0023 THEN LET J=J-100
0024 REM ** J-10
0025 IF HIT=12 THEN GO TO 4000
0026 GO TO 100
0027 IF LB=2 AND LC=2 AND LD=2 Y
0028 GO SUB 1000
0029 IF LB=1 OR LC=1 OR LD=1 THE
0030 GO SUB 1000
0031 IF LB=1 OR LC=1 THEN GO SUB
0032
0033 INPUT M: IF NOT M="H" AND
0034 M="T" THEN GO TO 1000
0035 IF M="T" THEN GO SUB 1000
0036 IF M="H" THEN GO SUB 1000
0037 PRINT "OK"
0038 LET X=X+1: Y=Y-1: C=C+1: D=D-1
0039 IF X=100 THEN LET J=J-100
0040 PRINT "INSTR 1: INK 1: F
0041 X+1: Y-1: C+1: D-1"
0042 IF X=100 THEN PRINT "INSTR
0043 2: INK 1: FLAG: BUNT X+1: Y-1:
0044
0045 REM ** J-10
0046 C=C+1: D=D-1
0047 IF HIT=12 THEN GO TO 4000
0048 GO TO 100
0049 LET X=INT (RND*10)+1: LET Y
0050 =INT (RND*10)+1
0051 IF LIX,Y1=48 THEN GO TO 10
0052
0053 IF LIX,Y1=48 AND LIX,Y-1
0054 AND LIX-1,Y1=48 AND LIX,Y
0055 THEN GO TO 1000
0056 LET X=X+1: Y=Y-1: C=C+1: D=D-1
0057 IF Y=0 THEN C=C+1: D=D-1
0058 PRINT "INSTR 1: INK 2: IN
0059 Y+1: C+1: D-1"
0060 IF X=100 THEN PRINT "INSTR
0061 2: INK 2: FLAG: BUNT X+1: Y-1:
0062
0063 RETURN
0064 LET X=X+1: Y=Y-1: C=C+1: D=D-1
0065 IF LB=1 OR LC=1 OR LD=1 THE
0066 GO TO 1000
0067 RETURN
0068 PRINT AT 14,14: PAPER 2: IN
YOUR CO*
0069 YOUR D4: FOR M=1 TO 2: FOR
0070 ST 10000: PAPER 2: INK 2: CH
0071 0: GO
0072 LET HOTS=HOTS+1
0073 IF HOTS=1 THEN GO SUB 1700
0074 IF HOTS=2 THEN GO SUB 1700
0075 IF HOTS=3 THEN GO SUB 1700
0076 RETURN
0077 LET J=CODE "C"
0078 LET L=L+1
0079 IF LB=2 THEN LET LB=3
0080 IF LB=1 THEN LET LB=2
0081 RETURN
0082 LET J=CODE "C"
0083 LET L=L+1
0084 IF LC=3 THEN LET LC=2: LET
0085
0086 IF LC=2 THEN LET LC=3
0087
0088 IF NOT Y=1 THEN GO TO 4000
0089 IF Y=1 THEN LET Y=Y+4: RETU
0090
0091 IF LIX,Y1=48 THEN LET Y=Y+4
0092 RETURN
0093 IF LIX,Y1=48 THEN LET LB=1:
0094 RETURN
0095 IF LB=1 THEN LET Y=Y+4: RET
0096
0097 LET X=X+1
0098 IF X=3 THEN LET X=X+4: RETU
0099
0100 IF LIX,Y1=48 THEN LET X=X+4
0101 RETURN
0102 IF LIX,Y1=48 THEN LET LB=1:
0103 RETURN
0104 IF LB=1 THEN LET Y=Y+4: RET
0105
0106 LET Y=Y-1
0107 IF Y=0 THEN LET Y=Y+4: RETU
0108
0109 IF LIX,Y1=48 THEN LET Y=Y+4
0110 RETURN
0111 IF LIX,Y1=48 THEN LET LB=1:
0112 RETURN
0113 PRINT AT 14,14: "YOU WON": G
0114 O TO 8040
0115 PRINT AT 14,14: "I WON": G
0116 O TO 8040
0117 PRINT AT 14,14: "AGAIN? (Y/N
0118
0119 INPUT Y
0120 IF NOT Y="Y" THEN GO TO 4000

```

```

1000 IF V#="Y" THEN RUN
1070 PRINT AT 21,14:"OK BYE": GO
    TO 9999
9000 REM == INITIALIZE ==
9010 FOR S=2000,100
9020 DIM A(10,10)
9030 DIM L(10,10)
9040 FOR S=1 TO 10: FOR U=1 TO 1
9050 LET R#H,M#L=143
9060 NEXT U: NEXT S
9070 RANDOMIZE
9080 LET P#100
9090 LET L#0: LET LC#0: LET LS#
9100 LET X#0: LET Y#0: LET Z#0:
9110 LET L#0: LET L#0: LET X#0:
9120 FOR S=1 TO 3: FOR Y#1 TO 3
9130 LET L(X,Y)=100: NEXT Y: NEXT
    S
9140 FOR S=1 TO 10: FOR Y#1 TO 1
9150 LET L(X,Y#40)=NEXT Y: NEXT
    S
9160 LET L(X,Y)=100: NEXT Y: NEXT
    S
9170 REM == GET UP SCORED ==
9180 FOR S=0 TO 5
9190 PRINT INK 1,0, PAPER 0, INK
    0,0:.....: INK 1, PAPER
    0,0
9200 NEXT S
9210 PRINT: INK 1:"ABCDEFGHIJ"
9220 REM ==
9230 FOR S=0 TO 9
9240 PRINT INK 1,0, PAPER 0, INK
    0,0:.....: INK 1,
    PAPER 0,0
9250 REM ==
9260 CLUT S#0: GO SUB 9260
9270 CLUT S#0: GO SUB 9270
9280 CLUT S#0: GO SUB 9280
9290 CLUT S#0: GO SUB 9290
9300 REM ==
9310 DRAW INK 0,70,0, DRAW INK 0
    0,70, DRAW INK 0,70, DRAW INK
    0,70, DRAW INK 0,70
9320 REM == MIDE SHIPS ==
9330 FOR S#0 TO 5: GO SUB 9330: FOR
    S#1 TO 4
9340 LET R#H,M#L=CODE "0"
9350 GO SUB 9340: NEXT P
9360 LET R#1 TO 5: GO SUB 9360: GO 0
    TO 9360 IF R#1 THEN GO TO 9360
9370 FOR P#1 TO 3
9380 LET R#H,M#L=CODE "0"
9390 GO SUB 9380: NEXT P: NEXT R
9400 LET R#0 TO 4
9410 LET R#0 TO 4: NEXT P: NEXT R
9420 LET R#0 TO 4: NEXT P: NEXT R
9430 LET R#0 TO 4: NEXT P: NEXT R
9440 LET R#0 TO 4: NEXT P: NEXT R
9450 LET R#0 TO 4: NEXT P: NEXT R
9460 LET R#0 TO 4: NEXT P: NEXT R
9470 LET R#0 TO 4: NEXT P: NEXT R
9480 LET R#0 TO 4: NEXT P: NEXT R
9490 LET R#0 TO 4: NEXT P: NEXT R
9500 LET R#0 TO 4: NEXT P: NEXT R
9510 LET R#0 TO 4: NEXT P: NEXT R
9520 LET R#0 TO 4: NEXT P: NEXT R
9530 LET R#0 TO 4: NEXT P: NEXT R
9540 LET R#0 TO 4: NEXT P: NEXT R
9550 LET R#0 TO 4: NEXT P: NEXT R
9560 LET R#0 TO 4: NEXT P: NEXT R
9570 LET R#0 TO 4: NEXT P: NEXT R
9580 LET R#0 TO 4: NEXT P: NEXT R
9590 LET R#0 TO 4: NEXT P: NEXT R
9600 LET R#0 TO 4: NEXT P: NEXT R
9610 LET R#0 TO 4: NEXT P: NEXT R
9620 LET R#0 TO 4: NEXT P: NEXT R
9630 LET R#0 TO 4: NEXT P: NEXT R
9640 LET R#0 TO 4: NEXT P: NEXT R
9650 LET R#0 TO 4: NEXT P: NEXT R
9660 LET R#0 TO 4: NEXT P: NEXT R
9670 LET R#0 TO 4: NEXT P: NEXT R
9680 LET R#0 TO 4: NEXT P: NEXT R
9690 LET R#0 TO 4: NEXT P: NEXT R
9700 LET R#0 TO 4: NEXT P: NEXT R
9710 LET R#0 TO 4: NEXT P: NEXT R
9720 LET R#0 TO 4: NEXT P: NEXT R
9730 LET R#0 TO 4: NEXT P: NEXT R
9740 LET R#0 TO 4: NEXT P: NEXT R
9750 LET R#0 TO 4: NEXT P: NEXT R
9760 LET R#0 TO 4: NEXT P: NEXT R
9770 LET R#0 TO 4: NEXT P: NEXT R
9780 LET R#0 TO 4: NEXT P: NEXT R
9790 LET R#0 TO 4: NEXT P: NEXT R
9800 LET R#0 TO 4: NEXT P: NEXT R
9810 LET R#0 TO 4: NEXT P: NEXT R
9820 LET R#0 TO 4: NEXT P: NEXT R
9830 LET R#0 TO 4: NEXT P: NEXT R
9840 LET R#0 TO 4: NEXT P: NEXT R
9850 LET R#0 TO 4: NEXT P: NEXT R
9860 LET R#0 TO 4: NEXT P: NEXT R
9870 LET R#0 TO 4: NEXT P: NEXT R
9880 LET R#0 TO 4: NEXT P: NEXT R
9890 LET R#0 TO 4: NEXT P: NEXT R
9900 LET R#0 TO 4: NEXT P: NEXT R
9910 LET R#0 TO 4: NEXT P: NEXT R
9920 LET R#0 TO 4: NEXT P: NEXT R
9930 LET R#0 TO 4: NEXT P: NEXT R
9940 LET R#0 TO 4: NEXT P: NEXT R
9950 LET R#0 TO 4: NEXT P: NEXT R
9960 LET R#0 TO 4: NEXT P: NEXT R
9970 LET R#0 TO 4: NEXT P: NEXT R
9980 LET R#0 TO 4: NEXT P: NEXT R
9990 LET R#0 TO 4: NEXT P: NEXT R

```

[illegible]



CMICALC (IBM Spectrum)

The best Spectrum spreadsheet you can buy written in machine code to be faster, to give you more space for data, and to allow more features to be included, it is guaranteed unobtainable.

- 80 columns or 280 rows.
- Fully prompted input.
- Extensive repeat facilities.
- Supports ALL Spectrum functions.
- Unique narrative facility.
- Tabular Sub-totals
- many more.

14 more programs share the quality of CMICALC than its might see more Spectrums in offices!"

"It is ideal for someone who has just found the spreadsheet concept, but it is also a very powerful tool for anyone who has used one previously" Sinclair User June 1983

Complete with comprehensive manual. **£29.95.**

ITIMCROSPHERE

72 ADDRESSABLE REMOTE-CONTROL MICRO-LEDS
TELEPHONE: 01-863 5901



WHEELIE (IBM Spectrum) Take the motorbike in your Spectrum for a bump-up. Bounce bumps, jumps and jumps as you seek the elusive ghostbike — and then your problems really start. Crash helmets available. **£5.95**

EVALUATION (IBM Spectrum) Penetration after the September. Prodigy, Spectrovision and some of the features on the fascinating journey from the past of life to the future. 88000000 and for the adventure take 3500 million years the last time it was played. Can you do it quicker? **— just £6.95**

CRASHES! NO PROBLEM! Any Spectrum Can you make a crash test car crash? Orbits and those all around you... in help a hungry rabbit in a rabbit world... 2 original games in one cassette **— only £4.95**

THE TRAIN GAME (Any Spectrum) The game that starts when model railways let off. Full graphics featuring passenger, goods & express trains, tunnels, stations, tunnels, tunnel games, train savengers, collisions, derailments, and everything else you expect from a major railway! **— Superb value £5.95**

QUARK 3D INVADERS

ORIG 1 48K

NEW!

Super 3D arcade action invaders will test your skill and 3D co-ordination to their limits. If hitting colour invaders scores of increasing difficulty will reward controlled 3D laser movement and fire sequences. Screen Required: 15.5K

DRAGON 32
 8800 05.50 • 8800 04.50 • 8800 04.50 • 8800 04.50
 8800 04.50 • 8800 04.50 • 8800 04.50 • 8800 04.50
 8800 04.50 • 8800 04.50 • 8800 04.50 • 8800 04.50

ORIG 1 48K
 8800 05.50 • 8800 04.50 • 8800 04.50 • 8800 04.50
 8800 04.50 • 8800 04.50 • 8800 04.50 • 8800 04.50

QUARK DATA P.O. BOX 51, Swindon, Wilt. Tel. 07933 40681.

Please supply

Price and postage only

Check for total amount enclosed

Name

Address

Mr. CHIP SOFTWARE

SPECTRUM GAMES

SPECTRUM DARTS (48K)

Five games of darts for 1-5 players, 800, Cricket, Killers, Round the board, Ninights and Crosses. Four levels of play per game. Take on the computer or friends at these games of skill and judgement. **£3.50**

WHEELER DEALER

As for the Commodore 64, but now available for the 48K Spectrum, Texas T190-44 and Dragon. **£5.50**

VIC-20

GAMES AND UTILITIES

JACKPOT

This is it, the ultimate Fruit Machine for the VIC with multiple, bold and crisp 100% machine code. "Overall Jackpot is a beautifully written simulation giving superb graphics, animation and use of colour. In fact, this program makes Commodore's Fruit Machine cartridge look unbelievably cheap and nasty." Home Computing Weekly No. 50 19/1/83 **£3.90**

KWAZE KWAKS

Accuracy and speed are required for this Shooting Gallery, superb use of colour and graphics in this new and challenging game from the author of Jackpot. 100% machine code, joystick or keyboard control. **£3.50**

PAUMANIA

Choose your own game from the following options — difficulty 1-3 speed 1-3, size of mine 1-3, visible or invisible mine, utilising moving power pills, define your own key controls, any combination, if this is your type of game, then this is the one for you, for the 1K expand VIC. **£2.50**

BUGSY (joystick Only)

This is a Minefield with a difference as you step on the mines whilst collecting purple boxes which give your both time and points, they disappear from beneath your feet. **DO NOT DESPAIR!** "BUGSY" will randomly replace the stones but avoid bumping into him or its sudden death! An original competitive and challenging game. **£3.50**

MINI-ROULETTE™ — PONTON — HELLO

Three great games of chance for the VIC, try to beat the computer, hours of fun, full colour, sound effects and more. **£3.50**

DATABASE — create your own files and records on tape **£7.50**

SUPER BANK MANAGER — A full feature version any memory size, but needs 1K expansion **£7.50**

COMMODORE 64 GAMES AND UTILITIES

JACKPOT 64

As last in these, specially written for the 64, by the author of "JACKPOT" the ultimate Fruit Machine program for the VIC. **£3.50**
Available from 1st Next

WESTMINSTER

A game for up to four players, can you lead the party of your choice and win the general election, you tour the 60 constituencies (total buying votes, when you can, just like the real thing), this must be one of the first board type games specially written for the computer. Why play on your own, have fun with your family and friends playing WESTMINSTER **£3.50**

RED ALERT

A game for 1-4 players, with sound and graphics. Make money in casinos, commit robbery, hide from the police, hire secret agents, (some of whom can be treacherous), negotiate for weapons, find and attack the secret nuclear base to launch the missile, and watch the havoc and destruction caused. There's no turning back from "RED ALERT" **£3.50**

WHEELER DEALER

A game for two to twenty players, become a system of the motor trade, you must obtain garbages, tyres and engines to produce cars for sale. Form syndicates, buy and exchange parts, buy dealerships, but be careful, you may become bankrupt and have to liquidate, find out what you are made of, have you got what it takes to become a WHEELER DEALER **£3.50**

LUNAR RESCUE

Our new version, avoid the asteroid belt, to rescue the stranded scientists, then fight your way back to the mother ship, fast reactions are required to safely land and dock your lunar module. **£3.50**

PURCHASE LEDGER

Easy to use, single entry, handles 400 invoices per month, Gross/Net purchases, V.A.T. **£14.50**
Disk Version **£17.00**
Available from 1st Next

SALES LEDGER — As Above **£14.50**
Disk Version **£17.00**

SALES JOURNAL

As above **£14.50**
Disk version **£17.00**

CHIPMON — Contains a one- or two-pass Assembler, Disassembler and Monitor. A programming aid for the development of machine code programs and routines on the CBM 64 **£12.50**

Other software available for the VIC and Commodore 64, send for free brochure, including RABBIT SOFTWARE at £3.50 each. Send Cheques/PO's to:

MR CHIP SOFTWARE

Dept P5, 1 NUTTLE PLACE, CLANDON, GUYNEED LE3 3BL.
Tel: 0492 49343

Wanted: High quality software, of all types, for export and UK distribution.
All programs now available on disc please allow £2.50 extra.

DEALER ENQUIRIES WELCOME



TOURIST TRAP

Get the holiday horrors in this spectacular board game for your 48K Spectrum.



This is a 'board game' program for two people to play against each other, rather than one person against the computer. It is a fairly large program using a major portion of the 48K RAM available on the larger Spectrum. This means that **SAVING** and **LOADING** will take quite some time. The playing of one game usually takes something like 40 minutes.

You can consider the program as operating in four main parts:

- 1) The initialisation of values and graphics, the printing of the instructions and then the playing board display.
- 2) The dice rolling routine to decide your throw.
- 3) The moving of your piece and the direction if required at that new position onto a subroutine which will decide your score for that throw.
- 4) The end game routine.

After the first part has been executed, lines 120 and 130 will keep the program looping through the second and third parts, automatically changing the players, until the end game condition is fulfilled to break out into the fourth part.

GOING FOURTH

The four sections operate as follows:

Part One Lines up to 100 call all the initialisation routines. Line 9000 allows the instructions to be displayed with the user-defined graphics are set up. These are the two playing pieces, plus a number of odd shaped 'blobs' used to create pictures in the course of play. The board printout begins at line 8000. The **PLOTDRAW** facility draws out a playing track around the outside of the display area allowing the centre to remain clear for the various graphics routines. The playing

squares are numbered using different combinations of **INK**, **PAPER** and **BRIGHTness**, these being controlled by the **DATA** statements at line 8000. It is the attributes on these squares that will decide the 'rate' of a player landing on them. It is *most* important that these are not altered or you will not be sent to the correct subroutines.

Part Two The dice routine is a straightforward affair. The variables for the 'pipe' are in line 8000 onward and it is printed out as required by **GOSUB 8425**. To roll the dice you press Enter and keep it pressed until you think you have the number you want. Releasing the pressure will stop the dice.

Part Three Moving the pieces around the board is more complicated. It is done by a series of **FOR...NEXT** statements. These take the value of the dice thrown as the second control value in the statement and move the piece along that number of squares. If that line runs out of squares then the next **FOR...NEXT** loop will use up the surplus. (I'm usually fond of conditional statements, but in this case found this system operated in a simpler manner.)

On arriving at a square, the attributes of the number of that square is read. If it is 104 then that turn ends, line 6900 changes over the player and your opponent will have their turn. If it is anything other than 104 then the value is converted to a **GOSUB** line and off you go to see what fate has in store.

The bulk of the program consists of these subroutines which I will let you discover for yourself. The operation of these routines is fairly

straightforward. There are nice things that will happen, not so nice things and, of course, some disasters — all of which lose or gain you points, lose you money and use up your petrol. If you run out of petrol you will have to pay for a rescue and if you are on the way to a location, you will lose the rest of that turn. The game ends when one person runs out of money... but the winner is the one with the most points, so towards the end of a game tactics can become important.

Fort Four Brief and simple. Once one person is out of money, line 150 fails to maintain the loop and the 'game over' signal shows. The final scores are then displayed. The score display routine is at line 8000 and is used at intervals throughout the game. Line 8000 ensures that petrol and money are only displayed to two decimal places; without this line you'll find that as the game progresses you end up with petrol designated in scientific notation!

EASY TO ENTER

Debugging a program of this size can be a bit daunting and so I would suggest it is entered in sections. Enter the program as listed but omitting all the subroutines from line 1000 to 6400. This will allow you to get the board display and the piece moving routines all in working order. To do this enter a temporary line, 6400 Return. RUNNING the program will then

result in everything happening except any of the scoring subroutines. The subroutines can then be entered one at a time and tested out. There is no need to keep going round the board to test these, just let the Spectrum print out the board, then break into the program and type in GOTO (the appropriate routine) and all will be well. (The 'Press enter to continue' sees the term 'INPUT:LINE 25' which means that the usual Break and Stop keys will not work. Never fear. Use Caps Shift and ' and you will get a 'stop in input' message.) The 'out of petrol' routine is the most difficult to test and may best be left until you are happy with the others.

Most of the graphics are straight off the keyboard. The only user-defined graphics that are vital are those for the two playing pieces and the dots on the dice. The others are not so vital but if you enter and RUN line 9000 to 9140 fairly early on, it is easy to see which keys are required.

WE'RE OFF...

When all is entered and you are ready to play you will list all you see the instructions, then the Spectrum will choose at random someone to start. The board is then printed out; the player whose turn it is will be shown on the right hand margin. If you throw the dice and land on a square that is not blue then some graphics and instructions will appear in the centre of the

screen. To continue to play; simply follow all the prompts. To enter in the spirit of it all you must imagine that you are on tour in a car, with a tent, and have the opportunity to visit various places... wild life parks... castles... the sea and so on.

So off you go. Happy Holiday.

VARIABLES

The main variables used are:

Cx	Cash on hand
D	Value of dice throw
Hx	Horizontal position of playing piece
Kx	Total score
LO	Flag indicating turn to be lost
M	Miles to travel
Px	Petrol in gallons
PL	Player
SUB	Subroutine in use
TI	Flag to check the number of times the petrol is evaluated
Vx	Vertical position of playing piece
Z	Line input

(x) will be one or two according to value of PL

The subroutines use many more variables, but they are confined within those routines and are not difficult to trace through. There are a number of DIMensioned strings and variables. They are used in several routines and are re-DIMensioned each time that routine is called.

Some sample screen illustrations from the program, Tourist Trap.

0	1	2	3	4	5	6	7	8	9	
29										1.0
26	IN THE				LEONE					1.1
27	RESCUE				ELEPHANTS					1.2
26	YOU SEE				MONKEYS					1.3
27	THESE				GIRAFFES					1.4
26	WHEELS				CAMELS					1.5
26	SCORE 2				SUPPLIES					1.6
26	FOR EACH				COLLIES					1.7
26					225000					1.8
24	23	22	21	20	19	18	17	16	15	1.9

0	1	2	3	4	5	6	7	8	9	
29										1.0
26										1.1
27										1.2
26										1.3
25										1.4
24	23	22	21	20	19	18	17	16	15	1.5

TOURIST TRAP

0	1	2	3	4	5	6	7	8	9	
20										10
25										11
27										12
28	A LUCKY DAY AT CASINO SUNDAYE AS LONG AS YOU CAN THROW DICE 3 DICES ENTER TO GO									13
30										14
34	23	22	21	20	19	18	17	16	15	

0	1	2	3	4	5	6	7	8	9	
24	MENU								10	
25	FRANKS								11	
26	PIZZA								12	
27	FRUIT PIE								13	
28	COFFEE								14	
29	PRESS ENTER TO FIND COST								15	
30	C1:0ICE								16	
31	32	33	34	35	36	37	38	39		

0	1	2	3	4	5	6	7	8	9	10
25	YOU ARE OUT OF PETROL.									10
26										11
27										12
28										13
29	YOU MUST PAY £10 + THROW OF GIDE FOR A TOW									14
30	31	32	33	34	35	36	37	38	39	40

0	1	2	3	4	5	6	7	8	9	10
25	YOU MAY BUY PETROL..YOU HAVE 0 GALLONS AT THE MOMENT									10
26	FULL TANK -10									11
27	FUEL 1 GALL..									12
28	CASH 2381.32									13
29										14
30	31	32	33	34	35	36	37	38	39	40

[illegible][illegible]

WATER



UPGRADE YOUR ZX SPECTRUM NOW!

The "CHEETAH" 32K Rampack simply plugs into the user port at the rear of your computer and increases the memory instantly to 48K.

- Fully compatible with all accessories via rear edge connector.
- No need to open computer and invalidate guarantee.
- Why send your computer away and wait weeks for upgrade.
- Fully cased, tested and guaranteed.

Why wait any longer?

ONLY **£39.95** inc. VAT and p&p



NOW MAKE YOUR SPECTRUM AND ZX81 TALK!

The CHEETAH "Sweet Talker" just plugs into the back of the computer using the existing power supply. Based on an allophone system you can easily program any word, sentence or phrase. Fully cased, tested, guaranteed and compatible with all accessories via rear edge connectors. Complete with demonstration cassette and full instructions. No more lonely nights!

Simply incredible at **£29.75**

Also available: Please quote when ordering whether Spectrum or ZX owner

16K Rampack for ZX81 **£19.75** 64K Rampack for ZX81 **£44.75**

Prices include VAT, Postage and Packing. Delivery normally 14 days. Export orders at no extra cost. Dealer enquiries welcome.

Send cheque/PO now to:

CHEETAH MARKETING LIMITED

Dept PS 2, 24 Ray Street, London EC1 Tel: 01-278 6954

Cheetah Products also available from branches of
Aumbelox

John Menzies 
and WHSMITH 

PLANET



This program began as an experiment in drawing three-dimensional shapes from different vantage points — a branch of geometry once of practical importance but now (along with much other geometric manipulation) on the decline in educational fashion. My aim was to compute and display successive views of an object as seen by a moving observer.

IN A FIX?

Let us take a fixed reference point, O , in space as the origin, and locate any point relative to O by three co-ordinates (x, y, z) representing its distance, East, North (the two-dimensional map reference) and in the vertical direction. The screen should be made to display a photograph of an object taken by an observer at co-ordinates, (l, m, n) . (To simplify matters, we assume that the vertical direction in space is taken to be at the top of the screen.)

Some algebraic manipulation showed that, if O is plotted at the centre of the screen (pixel 128,66), then the point with co-ordinates, $(1,0,0)$ should be plotted at $128+x1$, $66+y1$, co-ordinates $(0,1,0)$ at $128+x2$, $66+y2$, and co-ordinates $(0,0,1)$ at $128+x3$, $66+y3$; it should be noted that the displacements, $x1, \dots, y3$, are given by the

formulae in lines 340 to 360 of the program. These formulae include provision for the fact that the object appears smaller when viewed from further away.

With this information, an arbitrary point can be plotted — the point with co-ordinates (u, v, w) would appear at pixel, $128+u*x1+v*x2+w*x3$, $66+u*y1+v*y2+w*y3$.

LUCK OF THE DRAW

I decided to draw an octahedron. This figure has an interesting symmetrical shape which is extremely easy to draw. The corners, or vertices, are at the points $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ and their negatives. Line 360 of the program does the actual drawing.

You will see that the effect of line 360 is to draw the figure "without taking the pencil from the paper" — that is, only using the PLOT instruction once. That this is possible for the octahedron was another reason for my choice of solid. Such a figure is called Eulerian, after the mathematician Leonhard Euler. He showed that, in order to draw a figure without lifting your pencil, it is necessary and sufficient that each vertex lies on an even number of edges (in which case, you can start anywhere but you must finish at the start point), or exactly two don't lie on the edges (when

Here is an object lesson in geometry.

you must start at one of the edges and finish on another). It isn't necessary to demonstrate this — a routine for drawing an octahedron can be found by trial and error.

THE MECHANICS OF THE PROGRAM

The position co-ordinates of the observer are (l, m, n) and the velocity is (p, q, r) . This means that, in the time interval between one observation and another, l , m and n will change by amounts proportional to p , q and r respectively. This is implemented in line 400.

At first I kept the velocity fixed. However, if the observer is plotting a spaceship orbiting a planet, the velocity will also change. According to the laws of gravity discovered by Isaac Newton, the change of velocity will be in a direction towards the planet; that is, proportional to $(-l, -m, -n)/SQRT b$, where b is computed in line 340. The proportionality factor is the inverse square of the distance from the planet, is $1/b^2$.

At this point, I realised that there was a "game" element possible within my program, so I added controls, which change the velocity either towards or away from the planet. Unfortunately, the ship's engines are not very reliable and the actual change is somewhat reliable, but that is what games are all about, aren't they? The velocity change is implemented in line 410 and 430.

WHAT'S THE SCORE

The object of the game is to take survey photographs of the planet; these should be as large as possible so you'll have to get in real close. The score is determined by the size of the largest image to appear on the screen, with added bonuses for a very close approach. However, have I told you about the risk of crashing?

Since the controls only work

PLAY YOUR CARDS RIGHT

See if you can make this pontoon game stick on your Spectrum.



This program simulates the game of pontoon, in which you get to pit your wits against the computer.

When RUN, you will be dealt a card and asked to place a bet; bets can only be made up to a certain amount — within 10% of your credit. You will then be dealt another card and given the option to 'stick', 'twist' or 'buy'. This is done by pressing the appropriate key on the keyboard: the 'i' key to stick, the 't' key to twist and the 'b' key to buy.

YOUR DEAL...

If you've never played pontoon before, here is a brief résumé. You are dealt a single card, and on that you have to work out how good your hand might be and place a suitable bet accordingly. You then receive

your next card. The idea of the game is to make your cards up to a value of 21; King, Queens and Jacks are all worth 10 points each, and an Ace can be worth one or 11. With the two original cards you hold, you must decide whether to 'stick' which means you pick up no more cards, 'twist' which means

you receive another card (you can receive up to three more cards), or 'buy' which means you receive another card but you buy it from the dealer so that you effectively increase your bet.

The winner of a hand of pontoon is the one who gets their card to total 21 or the

AMT	— Amount of credit.
ITME	— The Spectrum or player dealing.
ACEI	— As ace has been dealt.
ME	— Value of the player's final hand.
V	— Current value of the cards.
T	— Number of cards dealt.
ODDS	— Calculation for the Spectrum to decide whether to stick or not.
Bet	— Amount bet on the initial card.
Buy	— Total amount for bought cards.
Stake	— The stake for each game.
SE	— The suit of each card.
RS	— The rank, Queen or King.

The notation used in this program.

TWIST AND SHOUT

When you have settled on your final hand and pressed the 'x' key to stick, then the computer deals its own hand, decides whether to stick or not and finally checks its hand against

Although there are copious REM statements throughout the program to help you follow its intricacies, a list of the variables used is given.

[illegible][illegible]

Personal Address: Spring '94

```

1170 REM 8
1180 PAPER 7: PRINT AT 1+3,2;"0"
1190 AT 1+4,3: CHRS 2: CHRS 3: CHRS 4
1200 AT 1+5,3: CHRS 5: CHRS 6: CHRS 7
1210 AT 1+6,3: CHRS 8: CHRS 9: CHRS 10
1220 AT 1+7,3: CHRS 11: CHRS 12: CHRS 13
1230 AT 1+8,3: CHRS 14: CHRS 15: CHRS 16
1240 AT 1+9,3: CHRS 17: CHRS 18: CHRS 19
1250 AT 1+10,3: CHRS 20: CHRS 21: CHRS 22
1260 AT 1+11,3: CHRS 23: CHRS 24: CHRS 25
1270 AT 1+12,3: CHRS 26: CHRS 27: CHRS 28
1280 AT 1+13,3: CHRS 29: CHRS 30: CHRS 31
1290 AT 1+14,3: CHRS 32: CHRS 33: CHRS 34
1300 AT 1+15,3: CHRS 35: CHRS 36: CHRS 37
1310 AT 1+16,3: CHRS 38: CHRS 39: CHRS 40
1320 AT 1+17,3: CHRS 41: CHRS 42: CHRS 43
1330 AT 1+18,3: CHRS 44: CHRS 45: CHRS 46
1340 AT 1+19,3: CHRS 47: CHRS 48: CHRS 49
1350 AT 1+20,3: CHRS 50: CHRS 51: CHRS 52
1360 AT 1+21,3: CHRS 53: CHRS 54: CHRS 55
1370 AT 1+22,3: CHRS 56: CHRS 57: CHRS 58
1380 AT 1+23,3: CHRS 59: CHRS 60: CHRS 61
1390 AT 1+24,3: CHRS 62: CHRS 63: CHRS 64
1400 AT 1+25,3: CHRS 65: CHRS 66: CHRS 67
1410 AT 1+26,3: CHRS 68: CHRS 69: CHRS 70
1420 AT 1+27,3: CHRS 71: CHRS 72: CHRS 73
1430 AT 1+28,3: CHRS 74: CHRS 75: CHRS 76
1440 AT 1+29,3: CHRS 77: CHRS 78: CHRS 79
1450 AT 1+30,3: CHRS 80: CHRS 81: CHRS 82
1460 AT 1+31,3: CHRS 83: CHRS 84: CHRS 85
1470 AT 1+32,3: CHRS 86: CHRS 87: CHRS 88
1480 AT 1+33,3: CHRS 89: CHRS 90: CHRS 91
1490 AT 1+34,3: CHRS 92: CHRS 93: CHRS 94
1500 AT 1+35,3: CHRS 95: CHRS 96: CHRS 97
1510 AT 1+36,3: CHRS 98: CHRS 99: CHRS 100
1520 AT 1+37,3: CHRS 101: CHRS 102: CHRS 103
1530 AT 1+38,3: CHRS 104: CHRS 105: CHRS 106
1540 AT 1+39,3: CHRS 107: CHRS 108: CHRS 109
1550 AT 1+40,3: CHRS 110: CHRS 111: CHRS 112
1560 AT 1+41,3: CHRS 113: CHRS 114: CHRS 115
1570 AT 1+42,3: CHRS 116: CHRS 117: CHRS 118
1580 AT 1+43,3: CHRS 119: CHRS 120: CHRS 121
1590 AT 1+44,3: CHRS 122: CHRS 123: CHRS 124
1600 AT 1+45,3: CHRS 125: CHRS 126: CHRS 127
1610 AT 1+46,3: CHRS 128: CHRS 129: CHRS 130
1620 AT 1+47,3: CHRS 131: CHRS 132: CHRS 133
1630 AT 1+48,3: CHRS 134: CHRS 135: CHRS 136
1640 AT 1+49,3: CHRS 137: CHRS 138: CHRS 139
1650 AT 1+50,3: CHRS 140: CHRS 141: CHRS 142
1660 AT 1+51,3: CHRS 143: CHRS 144: CHRS 145
1670 AT 1+52,3: CHRS 146: CHRS 147: CHRS 148
1680 AT 1+53,3: CHRS 149: CHRS 150: CHRS 151
1690 AT 1+54,3: CHRS 152: CHRS 153: CHRS 154
1700 AT 1+55,3: CHRS 155: CHRS 156: CHRS 157
1710 AT 1+56,3: CHRS 158: CHRS 159: CHRS 160
1720 AT 1+57,3: CHRS 161: CHRS 162: CHRS 163
1730 AT 1+58,3: CHRS 164: CHRS 165: CHRS 166
1740 AT 1+59,3: CHRS 167: CHRS 168: CHRS 169
1750 AT 1+60,3: CHRS 170: CHRS 171: CHRS 172
1760 AT 1+61,3: CHRS 173: CHRS 174: CHRS 175
1770 AT 1+62,3: CHRS 176: CHRS 177: CHRS 178
1780 AT 1+63,3: CHRS 179: CHRS 180: CHRS 181
1790 AT 1+64,3: CHRS 182: CHRS 183: CHRS 184
1800 AT 1+65,3: CHRS 185: CHRS 186: CHRS 187
1810 AT 1+66,3: CHRS 188: CHRS 189: CHRS 190
1820 AT 1+67,3: CHRS 191: CHRS 192: CHRS 193
1830 AT 1+68,3: CHRS 194: CHRS 195: CHRS 196
1840 AT 1+69,3: CHRS 197: CHRS 198: CHRS 199
1850 AT 1+70,3: CHRS 200: CHRS 201: CHRS 202
1860 AT 1+71,3: CHRS 203: CHRS 204: CHRS 205
1870 AT 1+72,3: CHRS 206: CHRS 207: CHRS 208
1880 AT 1+73,3: CHRS 209: CHRS 210: CHRS 211
1890 AT 1+74,3: CHRS 212: CHRS 213: CHRS 214
1900 AT 1+75,3: CHRS 215: CHRS 216: CHRS 217
1910 AT 1+76,3: CHRS 218: CHRS 219: CHRS 220
1920 AT 1+77,3: CHRS 221: CHRS 222: CHRS 223
1930 AT 1+78,3: CHRS 224: CHRS 225: CHRS 226
1940 AT 1+79,3: CHRS 227: CHRS 228: CHRS 229
1950 AT 1+80,3: CHRS 230: CHRS 231: CHRS 232
1960 AT 1+81,3: CHRS 233: CHRS 234: CHRS 235
1970 AT 1+82,3: CHRS 236: CHRS 237: CHRS 238
1980 AT 1+83,3: CHRS 239: CHRS 240: CHRS 241
1990 AT 1+84,3: CHRS 242: CHRS 243: CHRS 244
2000 AT 1+85,3: CHRS 245: CHRS 246: CHRS 247
2010 AT 1+86,3: CHRS 248: CHRS 249: CHRS 250
2020 AT 1+87,3: CHRS 251: CHRS 252: CHRS 253
2030 AT 1+88,3: CHRS 254: CHRS 255: CHRS 256
2040 AT 1+89,3: CHRS 257: CHRS 258: CHRS 259
2050 AT 1+90,3: CHRS 260: CHRS 261: CHRS 262
2060 AT 1+91,3: CHRS 263: CHRS 264: CHRS 265
2070 AT 1+92,3: CHRS 266: CHRS 267: CHRS 268
2080 AT 1+93,3: CHRS 269: CHRS 270: CHRS 271
2090 AT 1+94,3: CHRS 272: CHRS 273: CHRS 274
2100 AT 1+95,3: CHRS 275: CHRS 276: CHRS 277
2110 AT 1+96,3: CHRS 278: CHRS 279: CHRS 280
2120 AT 1+97,3: CHRS 281: CHRS 282: CHRS 283
2130 AT 1+98,3: CHRS 284: CHRS 285: CHRS 286
2140 AT 1+99,3: CHRS 287: CHRS 288: CHRS 289
2150 AT 1+100,3: CHRS 290: CHRS 291: CHRS 292
2160 AT 1+101,3: CHRS 293: CHRS 294: CHRS 295
2170 AT 1+102,3: CHRS 296: CHRS 297: CHRS 298
2180 AT 1+103,3: CHRS 299: CHRS 300: CHRS 301
2190 AT 1+104,3: CHRS 302: CHRS 303: CHRS 304
2200 AT 1+105,3: CHRS 305: CHRS 306: CHRS 307
2210 AT 1+106,3: CHRS 308: CHRS 309: CHRS 310
2220 AT 1+107,3: CHRS 311: CHRS 312: CHRS 313
2230 AT 1+108,3: CHRS 314: CHRS 315: CHRS 316
2240 AT 1+109,3: CHRS 317: CHRS 318: CHRS 319
2250 AT 1+110,3: CHRS 320: CHRS 321: CHRS 322
2260 AT 1+111,3: CHRS 323: CHRS 324: CHRS 325
2270 AT 1+112,3: CHRS 326: CHRS 327: CHRS 328
2280 AT 1+113,3: CHRS 329: CHRS 330: CHRS 331
2290 AT 1+114,3: CHRS 332: CHRS 333: CHRS 334
2300 AT 1+115,3: CHRS 335: CHRS 336: CHRS 337
2310 AT 1+116,3: CHRS 338: CHRS 339: CHRS 340
2320 AT 1+117,3: CHRS 341: CHRS 342: CHRS 343
2330 AT 1+118,3: CHRS 344: CHRS 345: CHRS 346
2340 AT 1+119,3: CHRS 347: CHRS 348: CHRS 349
2350 AT 1+120,3: CHRS 350: CHRS 351: CHRS 352
2360 AT 1+121,3: CHRS 353: CHRS 354: CHRS 355
2370 AT 1+122,3: CHRS 356: CHRS 357: CHRS 358
2380 AT 1+123,3: CHRS 359: CHRS 360: CHRS 361
2390 AT 1+124,3: CHRS 362: CHRS 363: CHRS 364
2400 AT 1+125,3: CHRS 365: CHRS 366: CHRS 367
2410 AT 1+126,3: CHRS 368: CHRS 369: CHRS 370
2420 AT 1+127,3: CHRS 371: CHRS 372: CHRS 373
2430 AT 1+128,3: CHRS 374: CHRS 375: CHRS 376
2440 AT 1+129,3: CHRS 377: CHRS 378: CHRS 379
2450 AT 1+130,3: CHRS 380: CHRS 381: CHRS 382
2460 AT 1+131,3: CHRS 383: CHRS 384: CHRS 385
2470 AT 1+132,3: CHRS 386: CHRS 387: CHRS 388
2480 AT 1+133,3: CHRS 389: CHRS 390: CHRS 391
2490 AT 1+134,3: CHRS 392: CHRS 393: CHRS 394
2500 AT 1+135,3: CHRS 395: CHRS 396: CHRS 397
2510 AT 1+136,3: CHRS 398: CHRS 399: CHRS 400
2520 AT 1+137,3: CHRS 401: CHRS 402: CHRS 403
2530 AT 1+138,3: CHRS 404: CHRS 405: CHRS 406
2540 AT 1+139,3: CHRS 407: CHRS 408: CHRS 409
2550 AT 1+140,3: CHRS 410: CHRS 411: CHRS 412
2560 AT 1+141,3: CHRS 413: CHRS 414: CHRS 415
2570 AT 1+142,3: CHRS 416: CHRS 417: CHRS 418
2580 AT 1+143,3: CHRS 419: CHRS 420: CHRS 421
2590 AT 1+144,3: CHRS 422: CHRS 423: CHRS 424
2600 AT 1+145,3: CHRS 425: CHRS 426: CHRS 427
2610 AT 1+146,3: CHRS 428: CHRS 429: CHRS 430
2620 AT 1+147,3: CHRS 431: CHRS 432: CHRS 433
2630 AT 1+148,3: CHRS 434: CHRS 435: CHRS 436
2640 AT 1+149,3: CHRS 437: CHRS 438: CHRS 439
2650 AT 1+150,3: CHRS 440: CHRS 441: CHRS 442
2660 AT 1+151,3: CHRS 443: CHRS 444: CHRS 445
2670 AT 1+152,3: CHRS 446: CHRS 447: CHRS 448
2680 AT 1+153,3: CHRS 449: CHRS 450: CHRS 451
2690 AT 1+154,3: CHRS 452: CHRS 453: CHRS 454
2700 AT 1+155,3: CHRS 455: CHRS 456: CHRS 457
2710 AT 1+156,3: CHRS 458: CHRS 459: CHRS 460
2720 AT 1+157,3: CHRS 461: CHRS 462: CHRS 463
2730 AT 1+158,3: CHRS 464: CHRS 465: CHRS 466
2740 AT 1+159,3: CHRS 467: CHRS 468: CHRS 469
2750 AT 1+160,3: CHRS 470: CHRS 471: CHRS 472
2760 AT 1+161,3: CHRS 473: CHRS 474: CHRS 475
2770 AT 1+162,3: CHRS 476: CHRS 477: CHRS 478
2780 AT 1+163,3: CHRS 479: CHRS 480: CHRS 481
2790 AT 1+164,3: CHRS 482: CHRS 483: CHRS 484
2800 AT 1+165,3: CHRS 485: CHRS 486: CHRS 487
2810 AT 1+166,3: CHRS 488: CHRS 489: CHRS 490
2820 AT 1+167,3: CHRS 491: CHRS 492: CHRS 493
2830 AT 1+168,3: CHRS 494: CHRS 495: CHRS 496
2840 AT 1+169,3: CHRS 497: CHRS 498: CHRS 499
2850 AT 1+170,3: CHRS 500: CHRS 501: CHRS 502
2860 AT 1+171,3: CHRS 503: CHRS 504: CHRS 505
2870 AT 1+172,3: CHRS 506: CHRS 507: CHRS 508
2880 AT 1+173,3: CHRS 509: CHRS 510: CHRS 511
2890 AT 1+174,3: CHRS 512: CHRS 513: CHRS 514
2900 AT 1+175,3: CHRS 515: CHRS 516: CHRS 517
2910 AT 1+176,3: CHRS 518: CHRS 519: CHRS 520
2920 AT 1+177,3: CHRS 521: CHRS 522: CHRS 523
2930 AT 1+178,3: CHRS 524: CHRS 525: CHRS 526
2940 AT 1+179,3: CHRS 527: CHRS 528: CHRS 529
2950 AT 1+180,3: CHRS 530: CHRS 531: CHRS 532
2960 AT 1+181,3: CHRS 533: CHRS 534: CHRS 535
2970 AT 1+182,3: CHRS 536: CHRS 537: CHRS 538
2980 AT 1+183,3: CHRS 539: CHRS 540: CHRS 541
2990 AT 1+184,3: CHRS 542: CHRS 543: CHRS 544
3000 AT 1+185,3: CHRS 545: CHRS 546: CHRS 547
3010 AT 1+186,3: CHRS 548: CHRS 549: CHRS 550
3020 AT 1+187,3: CHRS 551: CHRS 552: CHRS 553
3030 AT 1+188,3: CHRS 554: CHRS 555: CHRS 556
3040 AT 1+189,3: CHRS 557: CHRS 558: CHRS 559
3050 AT 1+190,3: CHRS 560: CHRS 561: CHRS 562
3060 AT 1+191,3: CHRS 563: CHRS 564: CHRS 565
3070 AT 1+192,3: CHRS 566: CHRS 567: CHRS 568
3080 AT 1+193,3: CHRS 569: CHRS 570: CHRS 571
3090 AT 1+194,3: CHRS 572: CHRS 573: CHRS 574
3100 AT 1+195,3: CHRS 575: CHRS 576: CHRS 577
3110 AT 1+196,3: CHRS 578: CHRS 579: CHRS 580
3120 AT 1+197,3: CHRS 581: CHRS 582: CHRS 583
3130 AT 1+198,3: CHRS 584: CHRS 585: CHRS 586
3140 AT 1+199,3: CHRS 587: CHRS 588: CHRS 589
3150 AT 1+200,3: CHRS 590: CHRS 591: CHRS 592
3160 AT 1+201,3: CHRS 593: CHRS 594: CHRS 595
3170 AT 1+202,3: CHRS 596: CHRS 597: CHRS 598
3180 AT 1+203,3: CHRS 599: CHRS 600: CHRS 601
3190 AT 1+204,3: CHRS 602: CHRS 603: CHRS 604
3200 AT 1+205,3: CHRS 605: CHRS 606: CHRS 607
3210 AT 1+206,3: CHRS 608: CHRS 609: CHRS 610
3220 AT 1+207,3: CHRS 611: CHRS 612: CHRS 613
3230 AT 1+208,3: CHRS 614: CHRS 615: CHRS 616
3240 AT 1+209,3: CHRS 617: CHRS 618: CHRS 619
3250 AT 1+210,3: CHRS 620: CHRS 621: CHRS 622
3260 AT 1+211,3: CHRS 623: CHRS 624: CHRS 625
3270 AT 1+212,3: CHRS 626: CHRS 627: CHRS 628
3280 AT 1+213,3: CHRS 629: CHRS 630: CHRS 631
3290 AT 1+214,3: CHRS 632: CHRS 633: CHRS 634
3300 AT 1+215,3: CHRS 635: CHRS 636: CHRS 637
3310 AT 1+216,3: CHRS 638: CHRS 639: CHRS 640
3320 AT 1+217,3: CHRS 641: CHRS 642: CHRS 643
3330 AT 1+218,3: CHRS 644: CHRS 645: CHRS 646
3340 AT 1+219,3: CHRS 647: CHRS 648: CHRS 649
3350 AT 1+220,3: CHRS 650: CHRS 651: CHRS 652
3360 AT 1+221,3: CHRS 653: CHRS 654: CHRS 655
3370 AT 1+222,3: CHRS 656: CHRS 657: CHRS 658
3380 AT 1+223,3: CHRS 659: CHRS 660: CHRS 661
3390 AT 1+224,3: CHRS 662: CHRS 663: CHRS 664
3400 AT 1+225,3: CHRS 665: CHRS 666: CHRS 667
3410 AT 1+226,3: CHRS 668: CHRS 669: CHRS 670
3420 AT 1+227,3: CHRS 671: CHRS 672: CHRS 673
3430 AT 1+228,3: CHRS 674: CHRS 675: CHRS 676
3440 AT 1+229,3: CHRS 677: CHRS 678: CHRS 679
3450 AT 1+230,3: CHRS 680: CHRS 681: CHRS 682
3460 AT 1+231,3: CHRS 683: CHRS 684: CHRS 685
3470 AT 1+232,3: CHRS 686: CHRS 687: CHRS 688
3480 AT 1+233,3: CHRS 689: CHRS 690: CHRS 691
3490 AT 1+234,3: CHRS 692: CHRS 693: CHRS 694
3500 AT 1+235,3: CHRS 695: CHRS 696: CHRS 697
3510 AT 1+236,3: CHRS 698: CHRS 699: CHRS 700
3520 AT 1+237,3: CHRS 701: CHRS 702: CHRS 703
3530 AT 1+238,3: CHRS 704: CHRS 705: CHRS 706
3540 AT 1+239,3: CHRS 707: CHRS 708: CHRS 709
3550 AT 1+240,3: CHRS 710: CHRS 711: CHRS 712
3560 AT 1+241,3: CHRS 713: CHRS 714: CHRS 715
3570 AT 1+242,3: CHRS 716: CHRS 717: CHRS 718
3580 AT 1+243,3: CHRS 719: CHRS 720: CHRS 721
3590 AT 1+244,3: CHRS 722: CHRS 723: CHRS 724
3600 AT 1+245,3: CHRS 725: CHRS 726: CHRS 727
3610 AT 1+246,3: CHRS 728: CHRS 729: CHRS 730
3620 AT 1+247,3: CHRS 731: CHRS 732: CHRS 733
3630 AT 1+248,3: CHRS 734: CHRS 735: CHRS 736
3640 AT 1+249,3: CHRS 737: CHRS 738: CHRS 739
3650 AT 1+250,3: CHRS 740: CHRS 741: CHRS 742
3660 AT 1+251,3: CHRS 743: CHRS 744: CHRS 745
3670 AT 1+252,3: CHRS 746: CHRS 747: CHRS 748
3680 AT 1+253,3: CHRS 749: CHRS 750: CHRS 751
3690 AT 1+254,3: CHRS 752: CHRS 753: CHRS 754
3700 AT 1+255,3: CHRS 755: CHRS 756: CHRS 757
3710 AT 1+256,3: CHRS 758: CHRS 759: CHRS 760
3720 AT 1+257,3: CHRS 761: CHRS 762: CHRS 763
3730 AT 1+258,3: CHRS 764: CHRS 765: CHRS 766
3740 AT 1+259,3: CHRS 767: CHRS 768: CHRS 769
3750 AT 1+260,3: CHRS 770: CHRS 771: CHRS 772
3760 AT 1+261,3: CHRS 773: CHRS 774: CHRS 775
3770 AT 1+262,3: CHRS 776: CHRS 777: CHRS 778
3780 AT 1+263,3: CHRS 779: CHRS 780: CHRS 781
3790 AT 1+264,3: CHRS 782: CHRS 783: CHRS 784
3800 AT 1+265,3: CHRS 785: CHRS 786: CHRS 787
3810 AT 1+266,3: CHRS 788: CHRS 789: CHRS 790
3820 AT 1+267,3: CHRS 791: CHRS 792: CHRS 793
3830 AT 1+268,3: CHRS 794: CHRS 795: CHRS 796
3840 AT 1+269,3: CHRS 797: CHRS 798: CHRS 799
3850 AT 1+270,3: CHRS 800: CHRS 801: CHRS 802
3860 AT 1+271,3: CHRS 803: CHRS 804: CHRS 805
3870 AT 1+272,3: CHRS 806: CHRS 807: CHRS 808
3880 AT 1+273,3: CHRS 809: CHRS 810: CHRS 811
3890 AT 1+274,3: CHRS 812: CHRS 813: CHRS 814
3900 AT 1+275,3: CHRS 815: CHRS 816: CHRS 817
3910 AT 1+276,3: CHRS 818: CHRS 819: CHRS 820
3920 AT 1+277,3: CHRS 821: CHRS 822: CHRS 823
3930 AT 1+278,3: CHRS 824: CHRS 825: CHRS 826
3940 AT 1+279,3: CHRS 827: CHRS 828: CHRS 829
3950 AT 1+280,3: CHRS 830: CHRS 831: CHRS 832
3960 AT 1+281,3: CHRS 833: CHRS 834: CHRS 835
3970 AT 1+282,3: CHRS 836: CHRS 837: CHRS 838
3980 AT 1+283,3: CHRS 839: CHRS 840: CHRS 841
3990 AT 1+284,3: CHRS 842: CHRS 843: CHRS 844
4000 AT 1+285,3: CHRS 845: CHRS 846: CHRS 847
4010 AT 1+286,3: CHRS 848: CHRS 849: CHRS 850
4020 AT 1+287,3: CHRS 851: CHRS 852: CHRS 853
4030 AT 1+288,3: CHRS 854: CHRS 855: CHRS 856
4040 AT 1+289,3: CHRS 857: CHRS 858: CHRS 859
4050 AT 1+290,3: CHRS 860: CHRS 861: CHRS 862
4060 AT 1+291,3: CHRS 863: CHRS 864: CHRS 865
4070 AT 1+292,3: CHRS 866: CHRS 867: CHRS 868
4080 AT 1+293,3: CHRS 869: CHRS 870: CHRS 871
4090 AT 1+294,3: CHRS 872: CHRS 873: CHRS 874
4100 AT 1+295,3: CHRS 875: CHRS 876: CHRS 877
4110 AT 1+296,3: CHRS 878: CHRS 879: CHRS 880
4120 AT 1+297,3: CHRS 881: CHRS 882: CHRS 883
4130 AT 1+298,3: CHRS 884: CHRS 885: CHRS 886
4140 AT 1+299,3: CHRS 887: CHRS 888: CHRS 889
4150 AT 1+300,3: CHRS 890: CHRS 891: CHRS 892
4160 AT 1+301,3: CHRS 893: CHRS 894: CHRS 895
4170 AT 1+302,3: CHRS 896: CHRS 897: CHRS 898
4180 AT 1+303,3: CHRS 899: CHRS 900: CHRS 901
4190 AT 1+304,3: CHRS 902: CHRS 903: CHRS 904
4200 AT 1+305,3: CHRS 905: CHRS 906: CHRS 907
4210 AT 1+306,3: CHRS 908: CHRS 909: CHRS 910
4220 AT 1+307,3: CHRS 911: CHRS 912: CHRS 913
4230 AT 1+308,3: CHRS 914: CHRS 915: CHRS 916
4240 AT 1+309,3: CHRS 917: CHRS 918: CHRS 919
4250 AT 1+310,3: CHRS 920: CHRS 921: CHRS 922
4260 AT 1+311,3: CHRS 923: CHRS 924: CHRS 925
4270 AT 1+312,3: CHRS 926: CHRS 927: CHRS 928
4280 AT 1+313,3: CHRS 929: CHRS 930: CHRS 931
4290 AT 1+314,3: CHRS 932: CHRS 933: CHRS 934
4300 AT 1+315,3: CHRS 935: CHRS 936: CHRS 937
4310 AT 1+316,3: CHRS 938: CHRS 939: CHRS 940
4320 AT 1+317,3: CHRS 941: CHRS 942: CHRS 943
4330 AT 1+318,3: CHRS 944: CHRS 945: CHRS 946
4340 AT 1+319,3: CHRS 947: CHRS 948: CHRS 949
4350 AT 1+320,3: CHRS 950: CHRS 951: CHRS 952
4360 AT 1+321,3: CHRS 953: CHRS 954: CHRS 955
4370 AT 1+322,3: CHRS 956: CHRS 957: CHRS 958
4380 AT 1+323,3: CHRS 959: CHRS 960: CHRS 961
4390 AT 1+324,3: CHRS 962: CHRS 963: CHRS 964
4400 AT 1+325,3: CHRS 965: CHRS 966: CHRS 967
4410 AT 1+326,3: CHRS 968: CHRS 969: CHRS 970
4420 AT 1+327,3: CHRS 971: CHRS 972: CHRS 973
4430 AT 1+328,3: CHRS 974: CHRS 975: CHRS 976
4440 AT 1+329,3: CHRS 977: CHRS 978: CHRS 979
4450 AT 1+330,3: CHRS 980: CHRS 981: CHRS 982
4460 AT 1+331,3: CHRS 983: CHRS 984: CHRS 985
4470 AT 1+332,3: CHRS 986: CHRS 987: CHRS 988
4480 AT 1+333,3: CHRS 989: CHRS 990: CHRS 991
4490 AT 1+334,3: CHRS 992: CHRS 993: CHRS 994
4500 AT 1+335,3: CHRS 995: CHRS 996: CHRS 997
4510 AT 1+336,3: CHRS 998: CHRS 999: CHRS 1000
4520 AT 1+337,3: CHRS 1001: CHRS 1002: CHRS 1003
4530 AT 1+338,3: CHRS 1004: CHRS 1005: CHRS 1006
4540 AT 1+339,3: CHRS 1007: CHRS 1008: CHRS 1009
4550 AT 1+340,3: CHRS 1010: CHRS 1011: CHRS 1012
4560 AT 1+341,3: CHRS 1013: CHRS 1014: CHRS 1015
4570 AT 1+342,3: CHRS 1016: CHRS 1017: CHRS 1018
4580 AT 1+343,3: CHRS 1019: CHRS 1020: CHRS 1021
4590 AT 1+344,3: CHRS 1022: CHRS 1023: CHRS 1024
4600 AT 1+345,3: CHRS 1025: CHRS 1026: CHRS 1027
4610 AT 1+346,3: CHRS 1028: CHRS 1029: CHRS 1030
4620 AT 1+347,3: CHRS 1031: CHRS 1032: CHRS 1033
4630 AT 1+348,3: CHRS 1034: CHRS 1035: CHRS 1036
4640 AT 1+349,3: CHRS 1037: CHRS 1038: CHRS 1039
4650 AT 1+350,3: CHRS 1040: CHRS 1041: CHRS 1042
4660 AT 1+351,3: CHRS 1043: CHRS 1044: CHRS 1045
4670 AT 1+352,3: CHRS 1046: CHRS 1047: CHRS 1048
4680 AT 1+353,3: CHRS 1049: CHRS 1050: CHRS 1051
4690 AT 1+354,3: CHRS 1052: CHRS 1053: CHRS 1054
4700 AT 1+355,3: CHRS 1055: CHRS 1056: CHRS 1057
4710 AT 1+356,3: CHRS 1058: CHRS 1059: CHRS 1060
4720 AT 1+357,3: CHRS 1061: CHRS 1062: CHRS 1063
4730 AT 1+358,3: CHRS 1064: CHRS 1065: CHRS 1066
4740 AT 1+
```

DEMOLITION

Get rid of your aggressive tendencies and have a smashing time on your Spectrum

In this game for your 16K Spectrum, the player is presented with a series of brick walls which slowly move up the screen. The walls must be demolished by dropping an explosive asterisk down onto

sections of the walls. The game is over when the wall reaches the top of the screen and the asterisk bumps into it.

The program has two features:

- The score is printed in the input area.

- The walls are activated by scrolling with SCR LT suppressed.

The directions for play are very

simple - once typed in and RUN, you simply press Enter to start the game and press the V key to drop the explosive asterisk as it moves across the top of the screen. The asterisk can move from right to left or vice versa; there is no warning given as to which side the asterisk will come.

Demolition is a simple, though very addictive game.



Photo courtesy of Games International Corp. GIG

1990 1991 1992 1993 1994 1995 1996 1997 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 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2

Source: *Journal of the American Statistical Association*, 1997, 92, 103-114.

100

[illegible]

DRAWING ON EXPERIENCE



There are a number of different methods to define characters on the Spectrum — some easy to understand, others totally incomprehensible. I saw one very interesting method POKEd, line by line. This is obviously not satisfactory when you have 31 characters to define. You would need 168 lines to POKE everything into the memory — that's lines 10 to 1680 using the conventional spacing system! There has to be a better method than that.

To cut down on the amount of typing we have to do, we can change the way we define our characters. To start with I will deal with the use of BIN. The Sinclair manual suggests that you use BIN when defining but I think this is a very long-winded way of doing it.

A binary number can be thought of as eight on/off switches each with a value; when the switch is on, it takes the value and when the switch is off, it becomes zero. The value of each switch is worked out from a power, ie $6/2$ is six to the power two, or 36.

For example, take a look at the binary number 0010001.

The value of BIN 0010001 is $0 + 0 + 36 + 16 + 0 + 4 + 0 + 1 = 57$. This can be seen fully illustrated in Fig. 1.

AND NEXT . . .

To turn a binary number into decimal all you need remember is the number in the top left-hand corner of the boxes so that you can add the 'on' ones together to get the result. Having converted all your BIN numbers into decimal you can now move onto the next part of this character generation method.

The Sinclair manual suggests you use a FOR . . . NEXT loop of zero to seven for some characters. I also use a FOR . . . NEXT loop, but I use it to cover all the characters to be defined, ie FOR A = USR "a" TO USR "Z" (Z is character to be defined) + 7. Inside the loop I READ all the decimal values and POKE them into the variable 'a'.

```
10 FOR A = USR "a"
    TO USR "a" + 7:
    REM this will
    require eight
    numbers in DATA
    statements
```

We hold tutorial on the subject of Spectrum graphics. Pull up a chair, plug in your computer and read on . . .

```
20 READ user: POKE
   a, user
30 NEXT A
40 DATA
   255,0,255,0,255,
   0,255,0
```

Try this program above. As you can see, the FOR . . . NEXT loop can be anything from one to 21 characters long. The DATA statement at the end contains the eight numbers required to create a striped character.

AT THE BAR

Enough said about user-definable graphics — what about programs which demonstrate high resolution graphics on the Spectrum? Well, here's a 3D Histogram program, which although not really very useful in any practical way, is a good demonstration of colour and Hi-res in action. The program surprisingly only uses two UGD characters, which have been POKEd in using the method shown previously.

QUICK ON THE DRAW

Once you've tried the Histogram program, you can move onto the next program. Sketch, which is a very sophisticated subprogram.

The controls for *Sketch* are shown in Fig. 2. The other controls you'll need to operate this program are:

- F — Flashing cursor.
- N — Draw mode cursor.
- D — Draw a line.
- M
- Move cursor.
- S — Save screen.
- V — CLR.
- C — Circle.
- O — Change INK colour.
- P — Change PAPER colour.
- B — Change BORDER colour.
- H — Help.

```

0000 PRINT AT 5.5; PAPER 1; " "
0005 " "
0010 NEXT a
0015 PRINT AT b.5; PAPER 5; INK
0020 "a"
0025 LET b=b+1
0030 PRINT AT 5.5; INK 1; PAPER
0035 "b"
0040 NEXT b
0045 INPUT "Press ENTER for next
0050 set RUN"; LINE 55: RUN
0055 STOP
0060 GOTO 54.175; DATA 5-175
0065 " "
0070 PRINT AT 5.5; PAPER 1; "a"
0075 " "
0080 " "
0085 " "
0090 " "
0095 " "
0100 " "
0105 " "
0110 " "
0115 " "
0120 " "
0125 " "
0130 " "
0135 " "
0140 " "
0145 " "
0150 " "
0155 " "
0160 " "
0165 " "
0170 " "
0175 " "
0180 " "
0185 " "
0190 " "
0195 " "
0200 " "
0205 " "
0210 " "
0215 " "
0220 " "
0225 " "
0230 " "
0235 " "
0240 " "
0245 " "
0250 " "
0255 " "
0260 " "
0265 " "
0270 " "
0275 " "
0280 " "
0285 " "
0290 " "
0295 " "
0300 " "
0305 " "
0310 " "
0315 " "
0320 " "
0325 " "
0330 " "
0335 " "
0340 " "
0345 " "
0350 " "
0355 " "
0360 " "
0365 " "
0370 " "
0375 " "
0380 " "
0385 " "
0390 " "
0395 " "
0400 " "
0405 " "
0410 " "
0415 " "
0420 " "
0425 " "
0430 " "
0435 " "
0440 " "
0445 " "
0450 " "
0455 " "
0460 " "
0465 " "
0470 " "
0475 " "
0480 " "
0485 " "
0490 " "
0495 " "
0500 " "
0505 " "
0510 " "
0515 " "
0520 " "
0525 " "
0530 " "
0535 " "
0540 " "
0545 " "
0550 " "
0555 " "
0560 " "
0565 " "
0570 " "
0575 " "
0580 " "
0585 " "
0590 " "
0595 " "
0600 " "
0605 " "
0610 " "
0615 " "
0620 " "
0625 " "
0630 " "
0635 " "
0640 " "
0645 " "
0650 " "
0655 " "
0660 " "
0665 " "
0670 " "
0675 " "
0680 " "
0685 " "
0690 " "
0695 " "
0700 " "
0705 " "
0710 " "
0715 " "
0720 " "
0725 " "
0730 " "
0735 " "
0740 " "
0745 " "
0750 " "
0755 " "
0760 " "
0765 " "
0770 " "
0775 " "
0780 " "
0785 " "
0790 " "
0795 " "
0800 " "
0805 " "
0810 " "
0815 " "
0820 " "
0825 " "
0830 " "
0835 " "
0840 " "
0845 " "
0850 " "
0855 " "
0860 " "
0865 " "
0870 " "
0875 " "
0880 " "
0885 " "
0890 " "
0895 " "
0900 " "
0905 " "
0910 " "
0915 " "
0920 " "
0925 " "
0930 " "
0935 " "
0940 " "
0945 " "
0950 " "
0955 " "
0960 " "
0965 " "
0970 " "
0975 " "
0980 " "
0985 " "
0990 " "
0995 " "
1000 " "
1005 " "
1010 " "
1015 " "
1020 " "
1025 " "
1030 " "
1035 " "
1040 " "
1045 " "
1050 " "
1055 " "
1060 " "
1065 " "
1070 " "
1075 " "
1080 " "
1085 " "
1090 " "
1095 " "
1100 " "
1105 " "
1110 " "
1115 " "
1120 " "
1125 " "
1130 " "
1135 " "
1140 " "
1145 " "
1150 " "
1155 " "
1160 " "
1165 " "
1170 " "
1175 " "
1180 " "
1185 " "
1190 " "
1195 " "
1200 " "
1205 " "
1210 " "
1215 " "
1220 " "
1225 " "
1230 " "
1235 " "
1240 " "
1245 " "
1250 " "
1255 " "
1260 " "
1265 " "
1270 " "
1275 " "
1280 " "
1285 " "
1290 " "
1295 " "
1300 " "
1305 " "
1310 " "
1315 " "
1320 " "
1325 " "
1330 " "
1335 " "
1340 " "
1345 " "
1350 " "
1355 " "
1360 " "
1365 " "
1370 " "
1375 " "
1380 " "
1385 " "
1390 " "
1395 " "
1400 " "
1405 " "
1410 " "
1415 " "
1420 " "
1425 " "
1430 " "
1435 " "
1440 " "
1445 " "
1450 " "
1455 " "
1460 " "
1465 " "
1470 " "
1475 " "
1480 " "
1485 " "
1490 " "
1495 " "
1500 " "
1505 " "
1510 " "
1515 " "
1520 " "
1525 " "
1530 " "
1535 " "
1540 " "
1545 " "
1550 " "
1555 " "
1560 " "
1565 " "
1570 " "
1575 " "
1580 " "
1585 " "
1590 " "
1595 " "
1600 " "
1605 " "
1610 " "
1615 " "
1620 " "
1625 " "
1630 " "
1635 " "
1640 " "
1645 " "
1650 " "
1655 " "
1660 " "
1665 " "
1670 " "
1675 " "
1680 " "
1685 " "
1690 " "
1695 " "
1700 " "
1705 " "
1710 " "
1715 " "
1720 " "
1725 " "
1730 " "
1735 " "
1740 " "
1745 " "
1750 " "
1755 " "
1760 " "
1765 " "
1770 " "
1775 " "
1780 " "
1785 " "
1790 " "
1795 " "
1800 " "
1805 " "
1810 " "
1815 " "
1820 " "
1825 " "
1830 " "
1835 " "
1840 " "
1845 " "
1850 " "
1855 " "
1860 " "
1865 " "
1870 " "
1875 " "
1880 " "
1885 " "
1890 " "
1895 " "
1900 " "
1905 " "
1910 " "
1915 " "
1920 " "
1925 " "
1930 " "
1935 " "
1940 " "
1945 " "
1950 " "
1955 " "
1960 " "
1965 " "
1970 " "
1975 " "
1980 " "
1985 " "
1990 " "
1995 " "
2000 " "
2005 " "
2010 " "
2015 " "
2020 " "
2025 " "
2030 " "
2035 " "
2040 " "
2045 " "
2050 " "
2055 " "
2060 " "
2065 " "
2070 " "
2075 " "
2080 " "
2085 " "
2090 " "
2095 " "
2100 " "
2105 " "
2110 " "
2115 " "
2120 " "
2125 " "
2130 " "
2135 " "
2140 " "
2145 " "
2150 " "
2155 " "
2160 " "
2165 " "
2170 " "
2175 " "
2180 " "
2185 " "
2190 " "
2195 " "
2200 " "
2205 " "
2210 " "
2215 " "
2220 " "
2225 " "
2230 " "
2235 " "
2240 " "
2245 " "
2250 " "
2255 " "
2260 " "
2265 " "
2270 " "
2275 " "
2280 " "
2285 " "
2290 " "
2295 " "
2300 " "
2305 " "
2310 " "
2315 " "
2320 " "
2325 " "
2330 " "
2335 " "
2340 " "
2345 " "
2350 " "
2355 " "
2360 " "
2365 " "
2370 " "
2375 " "
2380 " "
2385 " "
2390 " "
2395 " "
2400 " "
2405 " "
2410 " "
2415 " "
2420 " "
2425 " "
2430 " "
2435 " "
2440 " "
2445 " "
2450 " "
2455 " "
2460 " "
2465 " "
2470 " "
2475 " "
2480 " "
2485 " "
2490 " "
2495 " "
2500 " "
2505 " "
2510 " "
2515 " "
2520 " "
2525 " "
2530 " "
2535 " "
2540 " "
2545 " "
2550 " "
2555 " "
2560 " "
2565 " "
2570 " "
2575 " "
2580 " "
2585 " "
2590 " "
2595 " "
2600 " "
2605 " "
2610 " "
2615 " "
2620 " "
2625 " "
2630 " "
2635 " "
2640 " "
2645 " "
2650 " "
2655 " "
2660 " "
2665 " "
2670 " "
2675 " "
2680 " "
2685 " "
2690 " "
2695 " "
2700 " "
2705 " "
2710 " "
2715 " "
2720 " "
2725 " "
2730 " "
2735 " "
2740 " "
2745 " "
2750 " "
2755 " "
2760 " "
2765 " "
2770 " "
2775 " "
2780 " "
2785 " "
2790 " "
2795 " "
2800 " "
2805 " "
2810 " "
2815 " "
2820 " "
2825 " "
2830 " "
2835 " "
2840 " "
2845 " "
2850 " "
2855 " "
2860 " "
2865
```

$\frac{2\pi}{3}$	$\frac{\pi}{3}$	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{6}$	$\frac{\pi}{12}$	$\frac{\pi}{3}$	$\frac{\pi}{6}$
$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$

Fig. 1. An illustration to show that a binary number can be described as an object and still well-defined.



Fig. 1. The controls of movement in the Black Sea region.

```

0000  JUMP 10
0001  RETURN
0002  GOTO PAPER
0003  INPUT "Input new PAPER color"
0004  IF
0005  PAPER PA
0006  GOTO P711.7040
0007  PRINT AT 0.0: OVER 1: INK 0
0008  GOTO 1
0009  RETURN
0010  REM CIRCLE
0011  INPUT "CIRCLE (X,Y,R) X=":K
0012  IF
0013  Y="":GOTO 1
0014  X1=X1.0
0015  Y1=Y1.0
0016  REM BORDER
0017  REM BORDER
0018  INPUT "Input new BORDER color"
0019  IF
0020  BORDER LB
0021  GOTO 0.0
0022  RETURN
0023  GOTO 0.0
0024  PRINT AT 0.0:THREAT
0025  PRINT "CONTROL KEYS (INDICATED)"
0026  PRINT " 0 1 2 3"
0027  PRINT " 4 5 6 7"
0028  PRINT " 8 9 10 11"
0029  PRINT " 12 13 14 15"
0030  PRINT " 16 17 18 19"
0031  PRINT " 20 21 22 23"
0032  PRINT " 24 25 26 27"
0033  PRINT " 28 29 30 31"
0034  PRINT " 32 33 34 35"
0035  PRINT " 36 37 38 39"
0036  PRINT " 40 41 42 43"
0037  PRINT " 44 45 46 47"
0038  PRINT " 48 49 50 51"
0039  PRINT " 52 53 54 55"
0040  PRINT " 56 57 58 59"
0041  PRINT " 60 61 62 63"
0042  PRINT " 64 65 66 67"
0043  PRINT " 68 69 70 71"
0044  PRINT " 72 73 74 75"
0045  PRINT " 76 77 78 79"
0046  PRINT " 80 81 82 83"
0047  PRINT " 84 85 86 87"
0048  PRINT " 88 89 90 91"
0049  PRINT " 92 93 94 95"
0050  PRINT " 96 97 98 99"
0051  PRINT " 100 101 102 103"
0052  PRINT " 104 105 106 107"
0053  PRINT " 108 109 110 111"
0054  PRINT " 112 113 114 115"
0055  PRINT " 116 117 118 119"
0056  PRINT " 120 121 122 123"
0057  PRINT " 124 125 126 127"
0058  PRINT " 128 129 130 131"
0059  PRINT " 132 133 134 135"
0060  PRINT " 136 137 138 139"
0061  PRINT " 140 141 142 143"
0062  PRINT " 144 145 146 147"
0063  PRINT " 148 149 150 151"
0064  PRINT " 152 153 154 155"
0065  PRINT " 156 157 158 159"
0066  PRINT " 160 161 162 163"
0067  PRINT " 164 165 166 167"
0068  PRINT " 168 169 170 171"
0069  PRINT " 172 173 174 175"
0070  PRINT " 176 177 178 179"
0071  PRINT " 180 181 182 183"
0072  PRINT " 184 185 186 187"
0073  PRINT " 188 189 190 191"
0074  PRINT " 192 193 194 195"
0075  PRINT " 196 197 198 199"
0076  PRINT " 200 201 202 203"
0077  PRINT " 204 205 206 207"
0078  PRINT " 208 209 210 211"
0079  PRINT " 212 213 214 215"
0080  PRINT " 216 217 218 219"
0081  PRINT " 220 221 222 223"
0082  PRINT " 224 225 226 227"
0083  PRINT " 228 229 230 231"
0084  PRINT " 232 233 234 235"
0085  PRINT " 236 237 238 239"
0086  PRINT " 240 241 242 243"
0087  PRINT " 244 245 246 247"
0088  PRINT " 248 249 250 251"
0089  PRINT " 252 253 254 255"
0090  PRINT " 256 257 258 259"
0091  PRINT " 260 261 262 263"
0092  PRINT " 264 265 266 267"
0093  PRINT " 268 269 270 271"
0094  PRINT " 272 273 274 275"
0095  PRINT " 276 277 278 279"
0096  PRINT " 280 281 282 283"
0097  PRINT " 284 285 286 287"
0098  PRINT " 288 289 290 291"
0099  PRINT " 292 293 294 295"
0100  PRINT " 296 297 298 299"
0101  PRINT " 300 301 302 303"
0102  PRINT " 304 305 306 307"
0103  PRINT " 308 309 310 311"
0104  PRINT " 312 313 314 315"
0105  PRINT " 316 317 318 319"
0106  PRINT " 320 321 322 323"
0107  PRINT " 324 325 326 327"
0108  PRINT " 328 329 330 331"
0109  PRINT " 332 333 334 335"
0110  PRINT " 336 337 338 339"
0111  PRINT " 340 341 342 343"
0112  PRINT " 344 345 346 347"
0113  PRINT " 348 349 350 351"
0114  PRINT " 352 353 354 355"
0115  PRINT " 356 357 358 359"
0116  PRINT " 360 361 362 363"
0117  PRINT " 364 365 366 367"
0118  PRINT " 368 369 370 371"
0119  PRINT " 372 373 374 375"
0120  PRINT " 376 377 378 379"
0121  PRINT " 380 381 382 383"
0122  PRINT " 384 385 386 387"
0123  PRINT " 388 389 390 391"
0124  PRINT " 392 393 394 395"
0125  PRINT " 396 397 398 399"
0126  PRINT " 400 401 402 403"
0127  PRINT " 404 405 406 407"
0128  PRINT " 408 409 410 411"
0129  PRINT " 412 413 414 415"
0130  PRINT " 416 417 418 419"
0131  PRINT " 420 421 422 423"
0132  PRINT " 424 425 426 427"
0133  PRINT " 428 429 430 431"
0134  PRINT " 432 433 434 435"
0135  PRINT " 436 437 438 439"
0136  PRINT " 440 441 442 443"
0137  PRINT " 444 445 446 447"
0138  PRINT " 448 449 450 451"
0139  PRINT " 452 453 454 455"
0140  PRINT " 456 457 458 459"
0141  PRINT " 460 461 462 463"
0142  PRINT " 464 465 466 467"
0143  PRINT " 468 469 470 471"
0144  PRINT " 472 473 474 475"
0145  PRINT " 476 477 478 479"
0146  PRINT " 480 481 482 483"
0147  PRINT " 484 485 486 487"
0148  PRINT " 488 489 490 491"
0149  PRINT " 492 493 494 495"
0150  PRINT " 496 497 498 499"
0151  PRINT " 500 501 502 503"
0152  PRINT " 504 505 506 507"
0153  PRINT " 508 509 510 511"
0154  PRINT " 512 513 514 515"
0155  PRINT " 516 517 518 519"
0156  PRINT " 520 521 522 523"
0157  PRINT " 524 525 526 527"
0158  PRINT " 528 529 530 531"
0159  PRINT " 532 533 534 535"
0160  PRINT " 536 537 538 539"
0161  PRINT " 540 541 542 543"
0162  PRINT " 544 545 546 547"
0163  PRINT " 548 549 550 551"
0164  PRINT " 552 553 554 555"
0165  PRINT " 556 557 558 559"
0166  PRINT " 560 561 562 563"
0167  PRINT " 564 565 566 567"
0168  PRINT " 568 569 570 571"
0169  PRINT " 572 573 574 575"
0170  PRINT " 576 577 578 579"
0171  PRINT " 580 581 582 583"
0172  PRINT " 584 585 586 587"
0173  PRINT " 588 589 590 591"
0174  PRINT " 592 593 594 595"
0175  PRINT " 596 597 598 599"
0176  PRINT " 600 601 602 603"
0177  PRINT " 604 605 606 607"
0178  PRINT " 608 609 610 611"
0179  PRINT " 612 613 614 615"
0180  PRINT " 616 617 618 619"
0181  PRINT " 620 621 622 623"
0182  PRINT " 624 
```



STOP PRESS: NEW SPECTRUM DOMINOES 16K/48K £4.95

ZUCKMAN ZX81 (16K)

- * ALL MACHINE CODE (100%)
- * FOUR INDEPENDENT GHOSTS
- * HIGH-SCORE HALL OF FAME
- * AUTHENTIC ARCADE ACTION
- * TITLE/ODisplay MODES

ONLY £4.95 INC. P&P

*** ZX Spectrum** *Now* *Now* **FROGGY 16K or 48K**

SPYGLASS VERSION OF ARCADE GAME
WITH FULL ARCADE FEATURES

Excellent action arcade game
with 4 players, 4 lives each, 100%
machine code, authentic arcade
action, high score hall of fame
and title/odisplay modes.

Now available on the ZX Spectrum
in a new, exciting format.

Only £4.95 inc. P&P



FROGGY ZX81 (16K)

- * MOVING CARS, LOGS, TURTLES
- * ALLIGATORS, DRIVING TURTLES
- * FOUR 'SCREENS' OF ACTION
- * ALL ARCADE FEATURES
- * ENTIRELY MACHINE CODE

ONLY £4.95 INC. P&P



DJL SOFTWARE

DEPT/CE3, 9 TWEEDE CLOSE, SWINDON,
WILTS SN2 3PU Tel: 07931 724217

Trade enquiries welcome. Export Orders: Please add £1.00 per tape order.
Debit enquiries: Tiger Distribution, 1 Victoria Road, Walsley, Cheshire,
051 432 8888.

NAME _____

ADDRESS _____

Please send me the following _____

I understand the P.O. No. _____ for £ _____

ORDER 751 501 Software, from DJL, 1 Twede Close, Swindon, Wiltshire SN2 3PU

EARTH ATTACK



Written for the 16K and 48K Spectrum, this listing presents you with an exciting game which makes great use of machine code to save vital memory space. This program is divided into four main sections. These are:

- 1) Lines 1 to 180 contain all the initialisation and jumps to the user-defined graphics and machine code initialising routines. Since the amount of initialisation which needs to be done in this program is very large, there is quite a long pause after RUNING before the action starts - however, I shall explain later how this problem can be overcome.
- 2) Lines 200 to 999 contain the main game loop, including

its servant subroutines (scoring, 'smart bomb' and high score routines). This loop calls all the machine code routines, and ties together their responses and any other relevant BASIC needed for the 'action' of the game.

- 3) Lines 1000 to 2400 contain the data and POKING routines to set up first the user-defined graphics, then the machine code.
- 4) Lines 3000 to 3250 contain the three pages of instructions and game details which the computer cycles through between games.

The capital letters in quotes in the following lines should be entered in the Graphics mode, and when the program is RUN,

An exciting space adventure with just a sprinkling of machine code.

produce the corresponding user-defined graphics to that key:

140, 180, 200, 220, 245, 701, 715, 725, 9140.

There are numerous other variables used in the program, but they are mostly control or dummy variables which are only used once and their use is fairly self-explanatory.

WELCOME TO THE MACHINE

There are three main machine code routines used in the program. The data for them starts at line 2000, and I have disassembled them for you in Figs. 3a, 4a and 4b.

First is the 'SMART BOMB' routine. This is 26 bytes long, and starts at address 32400. Using the attributes file of the Spectrum display, it changes the PAPEN colour of every character position in the central third of the display (the part which the 'action' of this game utilises) to red. Whilst doing this it also calculates how many of the character positions previously contained yellow or their 16K colour. This is the number which it returns to the variable, a, when it returns to BASIC, which is then multiplied by 10 to increase your score according to how many alien 'your smart bomb destroyed'.

Next comes the 'SIDE SCROLL' and 'BUILD CITY' routines, which actually run together, and are called by USR 32425. The height that the skyscrapers are built up to depends upon the variable, b, which is POKED into location 32477. The routine operates once again only on the central third of the display. The reasons for this are:

- 1) This is an keeping with the character of the game.
- 2) it speeds up the workings of code which has to operate on the thousands of bytes in the Spectrum's D...File, and
- 3) due to what is termed the curious construction of the display file, it makes manipulating one third of the screen at a time a fairly simple task, but crossing between the thirds rather complicated.

The last of these three main routines is the 'CHECK FIRING' routine. Actually it does rather more than its name suggests. Its action is as follows.

First the laser line of the ship is drawn. At each step of this a check is made to see whether an alien has been hit or not.

If an alien has been hit, it checks whether this is a saucer or a monster, and adjusts the value of *bc* accordingly. It then creates a two stage explosion.

The routine then erases the laser line, and returns to BASIC.

IN SHORT . . .

There are two other shorter machine code routines also included. The routine which starts at address 3200 and shown in Fig. 3b is a servant subroutine for the 'CHECK FIRING' code. When called from within another routine, it prints a character to the screen by POKing the eight bytes which follow the call instruction to the character position, the start of which is specified by *hl*.

The routine starting at address 3200 produces the display of the city being destroyed at the end of the program, and is disassembled in Fig. 3c.

I mentioned before, the delay caused by the setting up of the machine code and user-defined graphics. Another problem with all this data is that it is an incredible waste of space, since once it has all been POKed into the relevant addresses when the program was first run, it just sits there

cluttering up your machine. And since the program as it stands uses very nearly all the space in a 65K machine, this meant that, should you be tempted, you would find it hard to squeeze in many more lines before the Spectrum gives up the ghost!

The answer is, of course, to SAVE the code as user-defined graphics in a separate file on tape, and re-LOAD them each time you LOAD the program

from tape. This is a very quick efficient method which really makes use of the space saving advantages which machine code can offer.

So, once your program has been typed in and you have RUN it once or twice to ensure that there are no errors in your typed-in version, you can erase every program line between 3000 and 3400 inclusive. Then type in the program lines shown in Fig. 1. Do not worry that

VARIABLES

The variables used, in order of their appearance, are:

- sf* - Initials of the highest score.
- hs* - The high score total.
- s* - Current score.
- bc* - $INT\ s/1000$ - used in calculating bonus bombs.
- n* - Used throughout the program as a control variable in FOR . . . NEXT loops.
- as* - Displays the number of ships remaining (initially four).
- i* - Number of laser shots.
- ab* - The number of smart bombs ready for use.
- x* and *y* - The co-ordinates of the ship.
- p* - The screen byte directly in front of the ship (top byte of the character position).
- rq* - The random variable used in selecting alien types.
- b* - The variable used for POKing into machine code (address 32470) to denote the height of the next building.
- a* - The dummy variable for jumps to machine code.
- p3* - Used in the high score routine to accept initials.

Fig. 3a. SMART BOMB disassembled.

```

M, M, ATTER FILE
ld c, 0
ld b, 255
LOOP   and hl
       ld a, hl
       cp b

```

```

jmp CHANGE
inc a
ld a, 8
ld (hl), a
djnz LOOP
ret
CHANGE add a, 16
       ld (hl), a
       djnz LOOP
       ret

```

Fig. 3b. CPOKE disassembled.

```

LOOP   ld b, 8
       pop de
       ld a, (de)
       inc de

```

```

push de
ld (hl), a
ld de, 256
add hl, de
djnz LOOP
ret

```

Fig. 3c. End of game display disassembled.

```

LOOP5  ld a, D-FILE
LOOP4  ld c, 52
       xor a

```

```

LOOP2  ld a,7
        ld hl,ATTR FILE
LOOP1  ld b,255
        ld hl,hl,a
        inc hl
        djnz LOOP1
        halt
        dec a
        jnz LOOP2
        ld b,1

```

Fig. 4a. 'NEW SCHOOL' disassembled.

```

SIDE   ld b,64
SCROLL: ld hl,D FILE
        ld de,32
        ld a,0
LOOP1  ld hl,hl,a
        add hl,de
        djnz LOOP1
        ld hl,ATTR FILE
        ld b,8
        ld de,32
        ld a,7
LOOP2  ld hl,hl,a
        add hl,de
        djnz LOOP2
        ld bc,2048
        ld de,D FILE
        ld hl,D FILE + 1
        ld b,255
        ld de,ATTR FILE

```

Fig. 4b. 'CHECK FIRING' disassembled.

```

        ld a,0
        ld hl,DFOC
        ld de,768
        add hl,de
        push hl
        ld b,5
        ld a,hl
LOOP1  op 129
        jr X1
        op 127
        jr X2
        ld hl,255
        inc hl
        djnz LOOP1
        halt
        halt
        halt
        jr X3
        inc c
        push bc
        pop hl,de
        push hl
        call CPOKE

```

```

LOOP3  ld a,hl
        srl a
        ld hl,hl,a
        dec hl
        dec c
        jnz LOOP3
        ld a,71
        op b
        jnz LOOP4
        djnz LOOP5
        ret

```

```

        ld hl,ATTR FILE
        + 1
        ld b,0
        ld c,15
        ld a,22
        rr 16
        ld a,c
        rr 16
        ld a,31
        rr 16
        ld a,16
        rr 16
        ld a,5
        rr 16
        ld a,17
        rr 16
        ld a,0
        rr 16
        ld a,148
        rr 16
        dec c
        djnz LOOP3

```

```

DEFB 0,65,36,24,
24,36,65,0
halt
halt
pop hl
push hl
call CPOKE
DEFB 153,90,
16,199,227,8,
90,153
halt
halt
pop hl
call CPOKE
DEFB 0,0,0,0,
0,0,0,0
pop bc
inc c
pop hl
ld a,5
ld hl,0
inc hl
inc b
op b
jnz LOOP2
ld b,0
ret

```

X3

LOOP2

some of these lines overwrite program lines that still exist - this is intended.

Now, find some room on a cassette somewhere, and SAVE the program and the two code files by typing:

RUN 2000

When re-LOADED, the program will automatically LOAD the two files and then RUN itself.

TIME FOR FUN!

And now, after all this effort, for the fun part - playing the game! Fig. 2 shows the keys used to control your rocket ship as it wings its way over a terrestrial cityscape. To add amusement there are a whole crop of little alien saucers out to get you. Although they don't actually shoot at you, there is a plentiful supply of these, and you can be destroyed in any of three ways: by crashing into them, by crashing into the city below you, or by accidentally moving into 'Typespace' above. In fact, your little ship is not very sturdy at all, but you do have one weapon on your side - your laser!

To start with you have 20 lasers, and your laser count is decremented by one each time you use them. You score 10 points for each saucer you hit, and 20 for each alien being (little yellow 'nutties' who stand on rooftops and are very hard to hit). Each time you bring the nose of your ship into contact with a green fuel base, your number of lasers is incremented by 20. It does not matter if you plough right through these fuel bases.

You get three smart bombs per 1,000 points scored. While the little bomb graphics are all shovelling beneath the city, each time you press the Space key the sky will turn red, and all visible aliens will be wiped out, and you will be given 10 points for each. Your ship will then accelerate out of the dangerous area.

The game ends when all five of your ships have been destroyed.

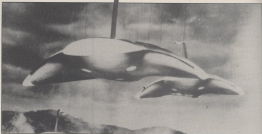


Fig.1

```

30 LOAD "EARTH" CODE
40 LOAD "EARTH" CODE
5000 SAVE "EARTH" CODE
5010 SAVE "EARTH" CODE
5020 SAVE "EARTH" CODE
5030 PRINT "SAVING FINISHED": 0
5040 RUN

```

Fig.2

EARTH ATTACK

```

USE KEYS: 1-S.....UP
           2-T.....DOWN
           3-R.....FIRE
           SHIFT.....EXTRA SPEED
           SPACE.....SHORT BOMBS
           PRESS "S" TO START

```

```

1 CLEAR 32768
2 DEEP 0:13 DEEP 0.24
3 GO TO 500
40 GO TO 500
50 BORDER 0
60 LET 0=0
70 LET 0=0
80 LET 0=0
90 LET 0=0
100 GO TO 5000
110 GO TO 500
120 LET 0=0
130 LET 0=0
140 LET 0=0
150 LET 0=0
160 LET 0=0
170 LET 0=0
180 LET 0=0
190 LET 0=0
200 LET 0=0
210 LET 0=0
220 LET 0=0
230 LET 0=0
240 LET 0=0
250 LET 0=0
260 LET 0=0
270 LET 0=0
280 LET 0=0
290 LET 0=0
300 LET 0=0
310 LET 0=0
320 LET 0=0
330 LET 0=0
340 LET 0=0
350 LET 0=0
360 LET 0=0
370 LET 0=0
380 LET 0=0
390 LET 0=0
400 LET 0=0
410 LET 0=0
420 LET 0=0
430 LET 0=0
440 LET 0=0
450 LET 0=0
460 LET 0=0
470 LET 0=0
480 LET 0=0
490 LET 0=0
500 LET 0=0
510 LET 0=0
520 LET 0=0
530 LET 0=0
540 LET 0=0
550 LET 0=0
560 LET 0=0
570 LET 0=0
580 LET 0=0
590 LET 0=0
600 LET 0=0
610 LET 0=0
620 LET 0=0
630 LET 0=0
640 LET 0=0
650 LET 0=0
660 LET 0=0
670 LET 0=0
680 LET 0=0
690 LET 0=0
700 LET 0=0
710 LET 0=0
720 LET 0=0
730 LET 0=0
740 LET 0=0
750 LET 0=0
760 LET 0=0
770 LET 0=0
780 LET 0=0
790 LET 0=0
800 LET 0=0
810 LET 0=0
820 LET 0=0
830 LET 0=0
840 LET 0=0
850 LET 0=0
860 LET 0=0
870 LET 0=0
880 LET 0=0
890 LET 0=0
900 LET 0=0
910 LET 0=0
920 LET 0=0
930 LET 0=0
940 LET 0=0
950 LET 0=0
960 LET 0=0
970 LET 0=0
980 LET 0=0
990 LET 0=0
1000 LET 0=0

```

```

340 IF IN 32768-355 THEN GO SUB
350 LET 0=0
360 LET 0=0
370 LET 0=0
380 LET 0=0
390 LET 0=0
400 LET 0=0
410 LET 0=0
420 LET 0=0
430 LET 0=0
440 LET 0=0
450 LET 0=0
460 LET 0=0
470 LET 0=0
480 LET 0=0
490 LET 0=0
500 LET 0=0
510 LET 0=0
520 LET 0=0
530 LET 0=0
540 LET 0=0
550 LET 0=0
560 LET 0=0
570 LET 0=0
580 LET 0=0
590 LET 0=0
600 LET 0=0
610 LET 0=0
620 LET 0=0
630 LET 0=0
640 LET 0=0
650 LET 0=0
660 LET 0=0
670 LET 0=0
680 LET 0=0
690 LET 0=0
700 LET 0=0
710 LET 0=0
720 LET 0=0
730 LET 0=0
740 LET 0=0
750 LET 0=0
760 LET 0=0
770 LET 0=0
780 LET 0=0
790 LET 0=0
800 LET 0=0
810 LET 0=0
820 LET 0=0
830 LET 0=0
840 LET 0=0
850 LET 0=0
860 LET 0=0
870 LET 0=0
880 LET 0=0
890 LET 0=0
900 LET 0=0
910 LET 0=0
920 LET 0=0
930 LET 0=0
940 LET 0=0
950 LET 0=0
960 LET 0=0
970 LET 0=0
980 LET 0=0
990 LET 0=0
1000 LET 0=0

```



```

0000 RETURN
0001 POINT AT 4.10, PLASM 1, ORN
0002 E OVER. FLASH 2: FOR 2=1 TO 3
0003 LET 2=0:20000:DEEP .1:0: NEXT
0004 2: PAUSE 2: PAUSE 1: IF 2=1 TH
0005 EN LET 2=0: DO SUB 200
0006 210 GO TO 2000
0007 220 GO TO 100
0008 LET 2=0:1
0009 IF 2=1:PRINT 10:1000:10: THEN LET
0010 2=2: PRINT AT 10.10, INK 2, PO
0011 2=0: "000"
0012 IF 2=2 THEN RETURN
0013 LET 1=1:10:1000:32400: DEEP
0014 0.1: DEEP .2:0: PRINT AT 20.2:
0015 2: PRINT AT 10.10=0: INK 2,
0016
0017 710 IF INT 0.1/1000:100 THEN LET
0018 2=2: PRINT AT 10.10, INK 2: PO
0019 2=0: "000"
0020 230 POINT 10:1000:
0021 240 FOR 2=0 TO 3: DO 200:77.1
0022 M: (200:31:1: LET 2=0:32400:
0023 250 POINT AT 2.0-1: PAPER 0: 2
0024 = 200: 0.1
0025 NEXT n: RETURN
0026 260 PAPER 2: INK 2: CLS
0027 270 POINT 0: "EARTH ATTACK"
0028
0029 LET 24="WELL DONE, YOU HAVE
0030 BEATEN THE HIGH SCORE, PLEASE
0031 ENTER YOUR INITIALS NOW"
0032 PRINT
0033 FOR 2=1 TO 10: LET 24=
0034 24: DEEP .002:0: NEXT 2
0035 PRINT: THE 2:
0036 LET 24=""
0037 PRINT: "FOR 2=1 TO 3:
0038 POINT 24:0: NEXT n
0039 290 FOR 2=1 TO 3
0040 300 POINT 24:0
0041 310 PAUSE 0: LET 24=INKEYS: IF
0042 LEN 24:1 THEN GO TO 270
0043 320 POINT 24:0: LET 24=24
0044 330 DEEP .1:0
0045 340 NEXT n
0046 350 PAUSE 20: RETURN
0047 360 STOP
0048 FOR 2=1 TO 5: DEEP 24
0049 1010 FOR 2=2 TO 5: DEEP 24
0050 1020 FOR 2=2 TO 5: DEEP 24
0051 1030 NEXT 2
0052 DATA 24 2: 210 1100000:24 1
0053 10000:24 1111000:24 117:24 1
0054 111000:24 110000:24 1110000
0055 1000 DATA 24 2: 1000:24 1001
0056 741:24 1002:24 1003:24 1004:24 1005:24 1006:24 1007:24 1008:24 1009:24 1010:24 1011:24 1012:24 1013:24 1014:24 1015:24 1016:24 1017:24 1018:24 1019:24 1020:24 1021:24 1022:24 1023:24 1024:24 1025:24 1026:24 1027:24 1028:24 1029:24 1030:24 1031:24 1032:24 1033:24 1034:24 1035:24 1036:24 1037:24 1038:24 1039:24 1040:24 1041:24 1042:24 1043:24 1044:24 1045:24 1046:24 1047:24 1048:24 1049:24 1050:24 1051:24 1052:24 1053:24 1054:24 1055:24 1056:24 1057:24 1058:24 1059:24 1060:24 1061:24 1062:24 1063:24 1064:24 1065:24 1066:24 1067:24 1068:24 1069:24 1070:24 1071:24 1072:24 1073:24 1074:24 1075:24 1076:24 1077:24 1078:24 1079:24 1080:24 1081:24 1082:24 1083:24 1084:24 1085:24 1086:24 1087:24 1088:24 1089:24 1090:24 1091:24 1092:24 1093:24 1094:24 1095:24 1096:24 1097:24 1098:24 1099:24 1100:24 1101:24 1102:24 1103:24 1104:24 1105:24 1106:24 1107:24 1108:24 1109:24 1110:24 1111:24 1112:24 1113:24 1114:24 1115:24 1116:24 1117:24 1118:24 1119:24 1120:24 1121:24 1122:24 1123:24 1124:24 1125:24 1126:24 1127:24 1128:24 1129:24 1130:24 1131:24 1132:24 1133:24 1134:24 1135:24 1136:24 1137:24 1138:24 1139:24 1140:24 1141:24 1142:24 1143:24 1144:24 1145:24 1146:24 1147:24 1148:24 1149:24 1150:24 1151:24 1152:24 1153:24 1154:24 1155:24 1156:24 1157:24 1158:24 1159:24 1160:24 1161:24 1162:24 1163:24 1164:24 1165:24 1166:24 1167:24 1168:24 1169:24 1170:24 1171:24 1172:24 1173:24 1174:24 1175:24 1176:24 1177:24 1178:24 1179:24 1180:24 1181:24 1182:24 1183:24 1184:24 1185:24 1186:24 1187:24 1188:24 1189:24 1190:24 1191:24 1192:24 1193:24 1194:24 1195:24 1196:24 1197:24 1198:24 1199:24 1200:24 1201:24 1202:24 1203:24 1204:24 1205:24 1206:24 1207:24 1208:24 1209:24 1210:24 1211:24 1212:24 1213:24 1214:24 1215:24 1216:24 1217:24 1218:24 1219:24 1220:24 1221:24 1222:24 1223:24 1224:24 1225:24 1226:24 1227:24 1228:24 1229:24 1230:24 1231:24 1232:24 1233:24 1234:24 1235:24 1236:24 1237:24 1238:24 1239:24 1240:24 1241:24 1242:24 1243:24 1244:24 1245:24 1246:24 1247:24 1248:24 1249:24 1250:24 1251:24 1252:24 1253:24 1254:24 1255:24 1256:24 1257:24 1258:24 1259:24 1260:24 1261:24 1262:24 1263:24 1264:24 1265:24 1266:24 1267:24 1268:24 1269:24 1270:24 1271:24 1272:24 1273:24 1274:24 1275:24 1276:24 1277:24 1278:24 1279:24 1280:24 1281:24 1282:24 1283:24 1284:24 1285:24 1286:24 1287:24 1288:24 1289:24 1290:24 1291:24 1292:24 1293:24 1294:24 1295:24 1296:24 1297:24 1298:24 1299:24 1300:24 1301:24 1302:24 1303:24 1304:24 1305:24 1306:24 1307:24 1308:24 1309:24 1310:24 1311:24 1312:24 1313:24 1314:24 1315:24 1316:24 1317:24 1318:24 1319:24 1320:24 1321:24 1322:24 1323:24 1324:24 1325:24 1326:24 1327:24 1328:24 1329:24 1330:24 1331:24 1332:24 1333:24 1334:24 1335:24 1336:24 1337:24 1338:24 1339:24 1340:24 1341:24 1342:24 1343:24 1344:24 1345:24 1346:24 1347:24 1348:24 1349:24 1350:24 1351:24 1352:24 1353:24 1354:24 1355:24 1356:24 1357:24 1358:24 1359:24 1360:24 1361:24 1362:24 1363:24 1364:24 1365:24 1366:24 1367:24 1368:24 1369:24 1370:24 1371:24 1372:24 1373:24 1374:24 1375:24 1376:24 1377:24 1378:24 1379:24 1380:24 1381:
```

TOOLBOX TRIO

Here are three BASIC utility programs.



The ability of the Spectrum to MERGE one program into another makes it very easy to tack one or more utility programs onto a BASIC program under development. Each utility program included in this article is a little longer than it need be because sensible prompts are included — however, these could be taken out if space is at a premium. Each program has been compacted to a reasonable number of lines so that it may be deleted fairly quickly when it is no longer required.

HEXADECIMAL/DECIMAL CONVERSION

Once you have entered the two programs, you use the following inputs to access the utilities:

RUN 9000 enters the Hex to decimal conversion routine.

RUN 9011 enters the decimal to Hex conversion routine.

The program will prompt for

the number to be converted. Should you wish to change the direction of conversion you do not need to break and re-RUN the other program, you can simply input 'T' and 123456. If you input 'Q', you stop the program.

PEEK AND POKE

RUN 9030 will PEEK a specified memory location. The program will ask you if you wish to read a single byte or a double byte, and then returns with a decimal number in the range 0 to 255 in the case of the former and 0 to 65535 in the case of the latter.

RUN 9030 will POKE a decimal number in the range 0 to 65535 into a specified memory location. Numbers greater than 255 will automatically be entered correctly into two consecutive memory locations.

ATTRIBUTE FINDER

This program will be of use to anyone investigating a BASIC

program when they only have a black and white television available. It may also be useful to anyone who has colour blindness. The BASIC program under investigation is stopped with the Break key and then RUN 9040 is entered.

Immediately the global attributes are displayed — the BORDER colour, the default PAPER colour, the default INK colour and BRIGHT/NORMAL mode — together with the attribute number that was set by the last PRINT statement before the break.

Then the user is prompted for a row and column co-ordinates until 'E' is entered, which will stop the program.

The Attribute Finder program itself obtains the global attributes by PEEKing three locations in the Spectrum workspace. This is done in line 9041. Local attributes are picked up using ATTR\$(1,C1) in line 9048.

The attribute number, held in N, is broken up into its component parts by the subroutine starting at line 9050. Notice the use of computed GOSUBs in lines 9052 and 9053 which pick up the names of the colours.



SPECTRUM BREAKOUT

Fancy a spot of demolition? Well, just read on...



Based on the ZX Spectrum, this game takes just over 6K of memory and makes full use of the sound and colour available on the machine.

The wall itself comprises three different coloured layers, each with 30 bricks. When 60 bricks have been knocked out, a new wall appears and the scores for each layer increases.

When you begin to play the game, you have five lives. However, each time you miss the ball with your bat, you will lose a life. The game ends when you have cleared all five walls or you have lost all five of your lives. All the appropriate sound effects have been included for when you hit the ball with your bat and when you knock out a brick from the wall.

When the game is not being played, the program cycles through a sequence of two explanation pages followed by a demonstration game in which the computer guides the bat. The highest score and the relevant player's name is displayed on screen at this time.

Here is a list of all the variables used in the program.

HS - Highest score.
MS - Name of the highest scorer.
F - General FOR...NEXT counter.
FP - Play flag (1 for the demonstration game).
P - X co-ordinate of the bat.
W - Wall number.
S - Score.
ES - Temporary INPUT string.
X,Y - Co-ordinates of the ball.
D - Direction of ball movement.
KDY - Movement vector for the ball.
FL - Flag for sound effects.

FT - Flag for hitting top of screen.
FA - Flag for hitting bat.
L - Lives.
A - Variable to indicate which layer has been hit.
I - Set to 0 for game, 8 for demonstration game.
C - Code of character at ball position.
H - Number of bricks knocked out.
P1 - Required position of bat during demonstration game.
D0 - Direction of bat movement during demonstration game.
YT - Temporary variable used in determining P1.

The listing for the program, Spectrum Breakout.

```
1 REM *****-OUT
2 SOUNDEN 0: LET HS=0: LET MS=""
3 GO SUB 20000:PRINT AT 21,2:
  10 PRINT AT 10,0: FLASH 1: SRI
  11 NT 1:PRESS ANY KEY TO PLAY-
  12 FOR I=1 TO 1000: IF INKEYS:
```

```
13 THEN GO TO 10
14 NEXT I: GO TO 2000
15 REPT 10,0: LET I=0: LET J=""
16
17 GO SUB 20000: GO TO 2000
18 REM *****-OUT
19 IF I=0 THEN GO TO 10
20 IF I=8 THEN GO TO 10
21 THEN GETIN: I=I AND INKEYS:
22 PRINT AT 21,0: "
```


STATISTICAL ANALYSIS

Check out your vital statistics with this great package.



This program, here presented for the 486 Spectrum (but with notes that follow for the program to be converted for the IBM model)

provides a thorough grounding in many of the statistical tests you may wish to use when analyzing quantities of data. Amongst the different

options of analysis you will have at your fingertips are standard deviation, the coefficient of variables, the standard error of mean, the F-test, the mode,


```

100 PRINT "TYPE IN YOUR TOTAL N  
NUMBER OF DATA PAIRS (n):" IN  
PUT "N: " CLS  
110 LET N=N: LET S=S+1  
120 PRINT INK 1: "TYPE IN YOUR X  
AND Y VALUES ENTER EACH ON  
A LINE"  
130 PRINT INK 3: "NOTE: IF YOU M  
AKE A MISTAKE WRITE IT DOWN  
AND YOU WILL BE ABLE TO ALT  
ER IT LATER."  
140 REM KIRI: DIM Y(1): DIM X(1)  
150 CLS  
160 PRINT: FOR N=1 TO N  
170 INPUT X: LET X(1)=X: LET N  
180 INPUT Y: LET Y(1)=Y: LET N  
190 PRINT INK 2: "S: (n):" "X:"  
200 REM "Y:"  
210 REM "S: (n):" POINT INK 2: "S:"  
220 REM "Y:"  
230 REM "S: (n):" POINT INK 2: "S:"  
240 REM "Y:"  
250 REM "S: (n):" POINT INK 2: "S:"  
260 REM "Y:"  
270 REM "S: (n):" POINT INK 2: "S:"  
280 REM "Y:"  
290 REM "S: (n):" POINT INK 2: "S:"  
300 REM "Y:"  
310 REM "S: (n):" POINT INK 2: "S:"  
320 REM "Y:"  
330 REM "S: (n):" POINT INK 2: "S:"  
340 REM "Y:"  
350 REM "S: (n):" POINT INK 2: "S:"  
360 REM "Y:"  
370 REM "S: (n):" POINT INK 2: "S:"  
380 REM "Y:"  
390 REM "S: (n):" POINT INK 2: "S:"  
400 REM "Y:"  
410 REM "S: (n):" POINT INK 2: "S:"  
420 REM "Y:"  
430 REM "S: (n):" POINT INK 2: "S:"  
440 REM "Y:"  
450 REM "S: (n):" POINT INK 2: "S:"  
460 REM "Y:"  
470 REM "S: (n):" POINT INK 2: "S:"  
480 REM "Y:"  
490 REM "S: (n):" POINT INK 2: "S:"  
500 REM "Y:"  
510 REM "S: (n):" POINT INK 2: "S:"  
520 REM "Y:"  
530 REM "S: (n):" POINT INK 2: "S:"  
540 REM "Y:"  
550 REM "S: (n):" POINT INK 2: "S:"  
560 REM "Y:"  
570 REM "S: (n):" POINT INK 2: "S:"  
580 REM "Y:"  
590 REM "S: (n):" POINT INK 2: "S:"  
600 REM "Y:"  
610 REM "S: (n):" POINT INK 2: "S:"  
620 REM "Y:"  
630 REM "S: (n):" POINT INK 2: "S:"  
640 REM "Y:"  
650 REM "S: (n):" POINT INK 2: "S:"  
660 REM "Y:"  
670 REM "S: (n):" POINT INK 2: "S:"  
680 REM "Y:"  
690 REM "S: (n):" POINT INK 2: "S:"  
700 REM "Y:"  
710 REM "S: (n):" POINT INK 2: "S:"  
720 REM "Y:"  
730 REM "S: (n):" POINT INK 2: "S:"  
740 REM "Y:"  
750 REM "S: (n):" POINT INK 2: "S:"  
760 REM "Y:"  
770 REM "S: (n):" POINT INK 2: "S:"  
780 REM "Y:"  
790 REM "S: (n):" POINT INK 2: "S:"  
800 REM "Y:"  
810 REM "S: (n):" POINT INK 2: "S:"  
820 REM "Y:"  
830 REM "S: (n):" POINT INK 2: "S:"  
840 REM "Y:"  
850 REM "S: (n):" POINT INK 2: "S:"  
860 REM "Y:"  
870 REM "S: (n):" POINT INK 2: "S:"  
880 REM "Y:"  
890 REM "S: (n):" POINT INK 2: "S:"  
900 REM "Y:"  
910 REM "S: (n):" POINT INK 2: "S:"  
920 REM "Y:"  
930 REM "S: (n):" POINT INK 2: "S:"  
940 REM "Y:"  
950 REM "S: (n):" POINT INK 2: "S:"  
960 REM "Y:"  
970 REM "S: (n):" POINT INK 2: "S:"  
980 REM "Y:"  
990 REM "S: (n):" POINT INK 2: "S:"  
1000 REM "Y:"  
1010 REM "S: (n):" POINT INK 2: "S:"  
1020 REM "Y:"  
1030 REM "S: (n):" POINT INK 2: "S:"  
1040 REM "Y:"  
1050 REM "S: (n):" POINT INK 2: "S:"  
1060 REM "Y:"  
1070 REM "S: (n):" POINT INK 2: "S:"  
1080 REM "Y:"  
1090 REM "S: (n):" POINT INK 2: "S:"  
1100 REM "Y:"  
1110 REM "S: (n):" POINT INK 2: "S:"  
1120 REM "Y:"  
1130 REM "S: (n):" POINT INK 2: "S:"  
1140 REM "Y:"  
1150 REM "S: (n):" POINT INK 2: "S:"  
1160 REM "Y:"  
1170 REM "S: (n):" POINT INK 2: "S:"  
1180 REM "Y:"  
1190 REM "S: (n):" POINT INK 2: "S:"  
1200 REM "Y:"  
1210 REM "S: (n):" POINT INK 2: "S:"  
1220 REM "Y:"  
1230 REM "S: (n):" POINT INK 2: "S:"  
1240 REM "Y:"  
1250 REM "S: (n):" POINT INK 2: "S:"  
1260 REM "Y:"  
1270 REM "S: (n):" POINT INK 2: "S:"  
1280 REM "Y:"  
1290 REM "S: (n):" POINT INK 2: "S:"  
1300 REM "Y:"  
1310 REM "S: (n):" POINT INK 2: "S:"  
1320 REM "Y:"  
1330 REM "S: (n):" POINT INK 2: "S:"  
1340 REM "Y:"  
1350 REM "S: (n):" POINT INK 2: "S:"  
1360 REM "Y:"  
1370 REM "S: (n):" POINT INK 2: "S:"  
1380 REM "Y:"  
1390 REM "S: (n):" POINT INK 2: "S:"  
1400 REM "Y:"  
1410 REM "S: (n):" POINT INK 2: "S:"  
1420 REM "Y:"  
1430 REM "S: (n):" POINT INK 2: "S:"  
1440 REM "Y:"  
1450 REM "S: (n):" POINT INK 2: "S:"  
1460 REM "Y:"  
1470 REM "S: (n):" POINT INK 2: "S:"  
1480 REM "Y:"  
1490 REM "S: (n):" POINT INK 2: "S:"  
1500 REM "Y:"  
1510 REM "S: (n):" POINT INK 2: "S:"  
1520 REM "Y:"  
1530 REM "S: (n):" POINT INK 2: "S:"  
1540 REM "Y:"  
1550 REM "S: (n):" POINT INK 2: "S:"  
1560 REM "Y:"  
1570 REM "S: (n):" POINT INK 2: "S:"  
1580 REM "Y:"  
1590 REM "S: (n):" POINT INK 2: "S:"  
1600 REM "Y:"  
1610 REM "S: (n):" POINT INK 2: "S:"  
1620 REM "Y:"  
1630 REM "S: (n):" POINT INK 2: "S:"  
1640 REM "Y:"  
1650 REM "S: (n):" POINT INK 2: "S:"  
1660 REM "Y:"  
1670 REM "S: (n):" POINT INK 2: "S:"  
1680 REM "Y:"  
1690 REM "S: (n):" POINT INK 2: "S:"  
1700 REM "Y:"  
1710 REM "S: (n):" POINT INK 2: "S:"  
1720 REM "Y:"  
1730 REM "S: (n):" POINT INK 2: "S:"  
1740 REM "Y:"  
1750 REM "S: (n):" POINT INK 2: "S:"  
1760 REM "Y:"  
1770 REM "S: (n):" POINT INK 2: "S:"  
1780 REM "Y:"  
1790 REM "S: (n):" POINT INK 2: "S:"  
1800 REM "Y:"  
1810 REM "S: (n):" POINT INK 2: "S:"  
1820 REM "Y:"  
1830 REM "S: (n):" POINT INK 2: "S:"  
1840 REM "Y:"  
1850 REM "S: (n):" POINT INK 2: "S:"  
1860 REM "Y:"  
1870 REM "S: (n):" POINT INK 2: "S:"  
1880 REM "Y:"  
1890 REM "S: (n):" POINT INK 2: "S:"  
1900 REM "Y:"  
1910 REM "S: (n):" POINT INK 2: "S:"  
1920 REM "Y:"  
1930 REM "S: (n):" POINT INK 2: "S:"  
1940 REM "Y:"  
1950 REM "S: (n):" POINT INK 2: "S:"  
1960 REM "Y:"  
1970 REM "S: (n):" POINT INK 2: "S:"  
1980 REM "Y:"  
1990 REM "S: (n):" POINT INK 2: "S:"  
2000 REM "Y:"  
2010 REM "S: (n):" POINT INK 2: "S:"  
2020 REM "Y:"  
2030 REM "S: (n):" POINT INK 2: "S:"  
2040 REM "Y:"  
2050 REM "S: (n):" POINT INK 2: "S:"  
2060 REM "Y:"  
2070 REM "S: (n):" POINT INK 2: "S:"  
2080 REM "Y:"  
2090 REM "S: (n):" POINT INK 2: "S:"  
2100 REM "Y:"  
2110 REM "S: (n):" POINT INK 2: "S:"  
2120 REM "Y:"  
2130 REM "S: (n):" POINT INK 2: "S:"  
2140 REM "Y:"  
2150 REM "S: (n):" POINT INK 2: "S:"  
2160 REM "Y:"  
2170 REM "S: (n):" POINT INK 2: "S:"  
2180 REM "Y:"  
2190 REM "S: (n):" POINT INK 2: "S:"  
2200 REM "Y:"  
2210 REM "S: (n):" POINT INK 2: "S:"  
2220 REM "Y:"  
2230 REM "S: (n):" POINT INK 2: "S:"  
2240 REM "Y:"  
2250 REM "S: (n):" POINT INK 2: "S:"  
2260 REM "Y:"  
2270 REM "S: (n):" POINT INK 2: "S:"  
2280 REM "Y:"  
2290 REM "S: (n):" POINT INK 2: "S:"  
2300 REM "Y:"  
2310 REM "S: (n):" POINT INK 2: "S:"  
2320 REM "Y:"  
2330 REM "S: (n):" POINT INK 2: "S:"  
2340 REM "Y:"  
2350 REM "S: (n):" POINT INK 2: "S:"  
2360 REM "Y:"  
2370 REM "S: (n):" POINT INK 2: "S:"  
2380 REM "Y:"  
2390 REM "S: (n):" POINT INK 2: "S:"  
2400 REM "Y:"  
2410 REM "S: (n):" POINT INK 2: "S:"  
2420 REM "Y:"  
2430 REM "S: (n):" POINT INK 2: "S:"  
2440 REM "Y:"  
2450 REM "S: (n):" POINT INK 2: "S:"  
24
```

```

900 LET H3=INT (H3+10+4+.5)/10
910 LET H4=R-1
920 POINT "H=";H3;"C=";C1;"
930 PRINT "Y=";Y1;"PRINT "S.D. OF Y="
940 PRINT "COEFF. OF V=";V1;"PRI
MT "S.E.M. FOR MEAN="
950 PRINT "SD OF Y=";S1;"PRINT "S.E.M.
COEFF. OF V=";V1
960 PRINT INK 3;"F-TEST" OVER
970 LET F=0
980 PRINT INK 1;"F-TEST TO COMPARE THE
PRECISION OF X WITH THAT OF
Y." PRINT "THE F VALUE=";F3;"PR
INT "FOR DF1 & DF2 OF "N1;"PRIN
T "THIS VALUE CAN BE LOOKED UP I
N F-TABLES TO FIND A PROBABILIT
Y IN VALUE FOR SIGNIFICANCE
TESTING."
990 GO SUB 600: GO TO 700
1000 REM *****
1010 STOP .0.30: PRINT INK 0;"OO
TH" OVER 1;RT 0.0
1020 INK 0; THEN GO TO 1000
1030 PRINT "DATA" OVER 1;RT 0
0
1040 PRINT "FOR X=1 TO X
1050 NEXT X: GO TO 1000
1060 FOR Y=1 TO Y
1070 PRINT "Y=";Y1;"PRINT "S.D.
OF Y="
1080 NEXT Y
1090 PRINT "GO SUB 600: GO TO 7
00
1100 REM *****
1110 PRINT "OVER" INK 0; FLASH
1120 "PLEASE WAIT" PRINT "OVER
1130 INK 1; FLASH 0;"THE COMPUTER
IS BANKING THE DATA" DATA PRI
OR LEFT
1140 FOR J=0 TO 1 STEP -.1: FOR K
1 TO 0
1150 LET L=0: IF B(K+1)=B(K) THE
N LET L=L+1: LET B(K)=B(K+1): L
ET B(K)=L
1160 LET A=0: IF C(K+1)=C(K) THE
N LET A=A+1: LET C(K)=C(K+1): L
ET C(K)=A
1170 NEXT K: PRINT AT 4,10;"
RT 4,10;J-1: PAUSE 0: PRINT AT
4,10;"RT 4,10;J-1: NEXT J
1180 CLS: STOP .0.30: PRINT INK
3;"BANKED DATA" OVER 1;RT 0.0
1190 FOR I=1 TO M
1200 PRINT "X=";X1;"PRINT "S.D.
OF X="
1210 NEXT X
1220 REM *****
1230 PRINT INK 0;"MIN. MAX. MEAN
AND" OVER 1;RT 0.0
1240 LET S=X/2: IF S=INT S THEN
GO TO 1260
1250 LET H1=S-10: LET H2=S+10: G
O TO 1240
1260 LET X=X/2: LET Y=Y/2: LET
H1=(H1+H2)/2: LET H2=(H1+H2)
C/2
1270 PRINT "MEDIAN OF X=";H1: DO
INT "MINIMUM VALUE OF X IS "
1280 PRINT "MAXIMUM VALUE OF X IS
"
1290 PRINT "MEDIAN OF Y=";H2: DO
INT "MINIMUM VALUE OF Y IS "
1300 PRINT "MAXIMUM VALUE OF Y IS
"
1310 LET H1=INT (H1+10+4+.5)/10: I
F H1=INT (H1+10+4+.5)/10 OR H1=1+1
0: LET S1=10+3/10 OR H1=1+1+1
0: S1=3/10 THEN PRINT "Median of
X approx. equal to mean of X
indicating normally distributed
X data."
1320 LET H3=INT (H3+10+4+.5)/10: I
F H3=INT (H3+10+4+.5)/10 OR H3=1+1
0: LET S3=10+3/10 OR H3=1+1+1
0: S3=3/10 THEN PRINT "Median of
Y approx. equal to mean of Y
indicating normally distributed
Y data."
1330 GO SUB 600: GO TO 700
1340 REM *****
1350 PRINT "OVER" INK 0; FLASH
1360 STOP .0.30: PRINT INK 0;"R
GRAPH OF X,Y" OVER 1;RT 0
1370 PRINT INK 1;"HAVE YOU BEEN
OUT OF HEADS" ROUTINE? (Y/N)
1380 INPUT IS: IF IS="N" THEN GO
TO 700
1390 IF NOT IS="Y" THEN CLS: GO
TO 700
1400 PRINT "TYPE IN THE NAME OF
THE X-Axis" INPUT X1
1410 PRINT "TYPE IN THE NAME OF
THE Y-Axis" INPUT Y1: CLS
1420 LET F=H1: LET S=C1:
1430 IF F=0 THEN LET F=0
1440 IF S=0 THEN LET S=0
1450 FOR Q=17 TO 100+20: PLOT 1
0.0: NEXT Q
1460 FOR H=0 TO 100+20: PLOT 1
0.0: NEXT H
1470 PRINT AT 0,1.3;H1: PRINT AT

```



```

OR 1:RT 0.0: " " PRINT
INR 1: "THIS IS THE MOST COMMONLY
OF OCCURRING VALUE IN ANY SET
OF DATA. IN THIS PROGRAM IT
T OF WILL BE SEEN IN THE L.I.S.
MINNED DATA IN NORMALLY
DISTRIBUTED DATA THE MODE
MEAN AND MEDIAN WOULD ALL BE
Y LEFT APPROXIMATELY EQUAL. AN
WOULD OR RIGHT SKEW IN THE DATA
MAKE THEM SIGNIFICANTLY
DIFFERENT FROM EACH OTHER
...
0000 GO SUB 000
0010 PRINT INR 0: "STATISTICAL AN
ALYSIS: " OVER 1:RT 0.0:
HERE CHOICES GIVE VARIOUS
CALCULATIONS ON THE DATA WHICH
WILL BE VERY TIME CONSUMING TO
DO MANUALLY. THEY ARE REGRESSION,
CORRELATION AND T-TEST.
0020 GO SUB 000
0030 PRINT INR 0: "REGRESSION ANAL
YSIS: " OVER 1:RT 0.0:
S IS AN ESTIMATE OF THE POSI
TION OF THE X DATA WITH THE
Y DATA. LINEAR REGRESSION ISUSE
D. WHERE IF THE ASSOCIATION IS
A STRAIGHT LINE THEN IT WILL HAVE
THE FORMULA Y=MX+K WHERE K=I
NTERCEPT ON THE Y-AXIS AND M=
REGRESSION COEFFICIENT (COR
RIDENT). IF THE ASSOCIATION IS
TOTALLY LINEAR THEN M WILL BE +1
OR -1 DEPENDING ON THE SLOPE OF
THE LINE. STATISTICALLY REG
RESSION IS USED TO DRAW A BEST
FIT LINE THROUGH SPREAD-OUT DATA.
0040 GO SUB 000
0050 PRINT INR 1: "HERE M IS CALCU
LATED AND USED TO FIND THE VALUE
OF K (BY K=C-(M*B) THE LINE IS TH
EN DRAWN FROM C THROUGH (B,C)
TO THE USER'S LIMITS OF THE
DATA. THIS LINE DRAWING METHOD
IS THE ONE USED ON THE PLOT A
GRAPH CHOICE. NOTE, THE LINE
ONLY WORKS FOR POSITIVELY COR
RELATED DATA. IF THE DATA IS
NEGATIVELY CORRELATED THE
LINE WILL RUN OFF THE SCREEN. TH
IS CAN BE AVOIDED IF PROGRAM LIN
ES 1700 & 1705 ARE DELETED. SEE M
ANUAL GRAPH INSTRUCTIONS.
...
0060 GO SUB 000
0070 PRINT INR 2: "CORRELATION CO
EFFICIENT: " OVER 1:RT 0.0: PRINT IN
C 1: "THIS IS ANOTHER EXPRESSION
OF THE ASSOCIATION BETWEEN THE
TWO SETS OF DATA. COMPLETE
CORRELATION WILL GIVE AN R
VALUE OF +1 OR -1 DEPENDING ON TH
SLOPE OF THE LINE. IF THE D
ATA IS TOTALLY UNCORRELATED THE
R WILL BE 0. CORRELATION AND
REGRESSION ANALYSIS ARE USU
ALLY USED IN CONJUNCTION AND ARE
OFTEN DISPLAYED ON THE GRAP
H OF DATA.
0080 GO SUB 000
0090 PRINT INR 3: "3-SAMPLE T-TEST
T: " OVER 1:RT 0.0:
ANY TYPES OF T-TEST ANALYSIS. T
HE ONE USED HERE IS FOR TWO SAM
PLES OF EQUAL LENGTH. THE TEST IS
USED TO DETERMINE WHETHER THE
DIFFERENCE BETWEEN THE TWO SETS OF DATA, TH
AT A VALUE IS CALCULATED
ALONG WITH THE DEGREES OF
FREEDOM (DF) FOR THE DATA. FROM
THESE, STUDENT'S T-TABLES CO
ULD BE USED TO FIND THE PROBABILITY
VALUE IF: FOR SIGNIFICANCE
TESTING: " OVER 1:RT 0.0:
THIS ROUTINE ALLOWS YOU
TO CHOOSE A BAR CHART OF X
OR Y VALUES. AGAIN IT IS LIM
ITED BY THE SCREEN SIZE OF THE
SPECTRUM THE VALUES ARE SORTED 1
STO 10 DOUBLE-WIDTH COLUMNS BY
A DIVISION FACTOR CALCULA
TED FROM THE MAXIMUM AND MINIMUM
DATA VALUES. THIS YOU MUST G
O THROUGH CHOICE NUMBER 3 (MEDIAN)
BEFORE USING THE BAR CHART. TH
E DATA IS SORTED INTO AN ARRAY AC
CORDING TO ITS PARTICULAR VALUE
- IF THE COLUMN DECREASES TOO LEAS
E > 10 THEN THE COMPUTED VAL
SCALE SHOWS ALL THE VALUES AS
REQUIRED. THE USE OF THIS IS THAT
IT WILL SHOW YOU IF THE DATA IS
NONNALLY DISTRIBUTED THIS IS RECO
GNIZED BY MANY STATS TESTS.
0100 GO SUB 000
0110 PRINT "PRESS F TO READ THESE
INSTRUCTIONS AGAIN
C TO COPY, AND ENTER TO RETURN
N TO THE MAIN PROGRAM: "
INPUT M
0120 IF M="F" THEN CLS : GO TO
0000
0130 IF M="C" THEN CLS : GO TO 7
0140 IF M="E" THEN COPY : CLS :
GO TO 0700
0150 GOTO 0700
0160 GOTO 0700
0170 GOTO 0700
0180 GOTO 0700
0190 GOTO 0700
0200 GOTO 0700
0210 GOTO 0700
0220 GOTO 0700
0230 GOTO 0700
0240 GOTO 0700
0250 GOTO 0700
0260 GOTO 0700
0270 GOTO 0700
0280 GOTO 0700
0290 GOTO 0700
0300 GOTO 0700
0310 GOTO 0700
0320 GOTO 0700
0330 GOTO 0700
0340 GOTO 0700
0350 GOTO 0700
0360 GOTO 0700
0370 GOTO 0700
0380 GOTO 0700
0390 GOTO 0700
0400 GOTO 0700
0410 GOTO 0700
0420 GOTO 0700
0430 GOTO 0700
0440 GOTO 0700
0450 GOTO 0700
0460 GOTO 0700
0470 GOTO 0700
0480 GOTO 0700
0490 GOTO 0700
0500 GOTO 0700
0510 GOTO 0700
0520 GOTO 0700
0530 GOTO 0700
0540 GOTO 0700
0550 GOTO 0700
0560 GOTO 0700
0570 GOTO 0700
0580 GOTO 0700
0590 GOTO 0700
0600 GOTO 0700
0610 GOTO 0700
0620 GOTO 0700
0630 GOTO 0700
0640 GOTO 0700
0650 GOTO 0700
0660 GOTO 0700
0670 GOTO 0700
0680 GOTO 0700
0690 GOTO 0700
0700 GOTO 0700
0710 GOTO 0700
0720 GOTO 0700
0730 GOTO 0700
0740 GOTO 0700
0750 GOTO 0700
0760 GOTO 0700
0770 GOTO 0700
0780 GOTO 0700
0790 GOTO 0700
0800 GOTO 0700
0810 GOTO 0700
0820 GOTO 0700
0830 GOTO 0700
0840 GOTO 0700
0850 GOTO 0700
0860 GOTO 0700
0870 GOTO 0700
0880 GOTO 0700
0890 GOTO 0700
0900 GOTO 0700
0910 GOTO 0700
0920 GOTO 0700
0930 GOTO 0700
0940 GOTO 0700
0950 GOTO 0700
0960 GOTO 0700
0970 GOTO 0700
0980 GOTO 0700
0990 GOTO 0700

```

MORSE TRAINER

Use your Spectrum to
learn the Morse Code.

The program itself is a fairly straightforward listing being very user friendly and menu driven. In the test mode, the instructions for operation are displayed and a Morse character is BEEPed. It is then up to the user to INPUT a letter. Should you not provide the correct letter, you will be informed. You are given three opportunities to give the correct answer, after which the correct letter is displayed along with its relevant dots and dashes, and the letter is BEEPed in Morse code again.

A LOAD OF CODE

Mr Gilbert has used a number of useful POKEs in this program. Here is a brief explanation of these POKEs.

POKE 23609,X — This gives the

keyboard click varying duration. When X=0, you hear the standard click. However, the value of X can be between 0 and 255, but when numbers greater than 10 are used, even though the feedback beep is more noticeable, there is also a noticeable difference in the speed of the auto-repeat facility. This can be frustrating when editing long program lines. It also does not work with INKEYs.

POKE 20688,X — When this location is POKEd with a zero, it disengages the Caps Shift lock. When POKEd with an eight, it engages the Caps Shift lock. This can be very convenient as it saves checking for capital or lower case letters when a user is INPUTing information to a program.

POKE 23692, 1 — This will ensure that the program will not stop with the message:

"Scroll?..." In this case, the PRINTing is started at co-ordinates 21,0 which would normally present you with that smacking little two letter word. Try omitting the POKE 23692 line and have a look what happens!

Is the FOR b NEXT b loop, it is attached at the first letter and is turned into its Morse equivalent by READING the DATA the requisite number of times as set in line 95.

Once you get used to the program, the PAUSE times can be easily changed if you begin to find them a bit slow. Ex 2381 users who are dubious of utilizing the PAUSE instruction due to the twitching screen syndrome need have no fears.

```

1 REM MORSE TRAINER
2 REM © M.Gilbert MCHL:00011
3 POKE 23609,0
4 DIM M$(255)
5 DIM L$(255)
6 INPUT "ENTER LETTER:" L$
7 INPUT "ENTER DOTS/DASHES:" M$
8 LET N=LEN(M$)
9 FOR I=1 TO N
10   LET C=M$(I)
11   IF C="." THEN GO TO 12
12   IF C="-" THEN GO TO 13
13   IF C=" " THEN GO TO 14
14   IF C="X" THEN GO TO 15
15   IF C="Y" THEN GO TO 16
16   IF C="Z" THEN GO TO 17
17   IF C="A" THEN GO TO 18
18   IF C="B" THEN GO TO 19
19   IF C="C" THEN GO TO 20
20   IF C="D" THEN GO TO 21
21   IF C="E" THEN GO TO 22
22   IF C="F" THEN GO TO 23
23   IF C="G" THEN GO TO 24
24   IF C="H" THEN GO TO 25
25   IF C="I" THEN GO TO 26
26   IF C="J" THEN GO TO 27
27   IF C="K" THEN GO TO 28
28   IF C="L" THEN GO TO 29
29   IF C="M" THEN GO TO 30
30   IF C="N" THEN GO TO 31
31   IF C="O" THEN GO TO 32
32   IF C="P" THEN GO TO 33
33   IF C="Q" THEN GO TO 34
34   IF C="R" THEN GO TO 35
35   IF C="S" THEN GO TO 36
36   IF C="T" THEN GO TO 37
37   IF C="U" THEN GO TO 38
38   IF C="V" THEN GO TO 39
39   IF C="W" THEN GO TO 40
40   IF C="X" THEN GO TO 41
41   IF C="Y" THEN GO TO 42
42   IF C="Z" THEN GO TO 43
43   IF C=" " THEN GO TO 44
44   IF C="." THEN GO TO 45
45   IF C="-" THEN GO TO 46
46   IF C=" " THEN GO TO 47
47   IF C="X" THEN GO TO 48
48   IF C="Y" THEN GO TO 49
49   IF C="Z" THEN GO TO 50
50   IF C="A" THEN GO TO 51
51   IF C="B" THEN GO TO 52
52   IF C="C" THEN GO TO 53
53   IF C="D" THEN GO TO 54
54   IF C="E" THEN GO TO 55
55   IF C="F" THEN GO TO 56
56   IF C="G" THEN GO TO 57
57   IF C="H" THEN GO TO 58
58   IF C="I" THEN GO TO 59
59   IF C="J" THEN GO TO 60
60   IF C="K" THEN GO TO 61
61   IF C="L" THEN GO TO 62
62   IF C="M" THEN GO TO 63
63   IF C="N" THEN GO TO 64
64   IF C="O" THEN GO TO 65
65   IF C="P" THEN GO TO 66
66   IF C="Q" THEN GO TO 67
67   IF C="R" THEN GO TO 68
68   IF C="S" THEN GO TO 69
69   IF C="T" THEN GO TO 70
70   IF C="U" THEN GO TO 71
71   IF C="V" THEN GO TO 72
72   IF C="W" THEN GO TO 73
73   IF C="X" THEN GO TO 74
74   IF C="Y" THEN GO TO 75
75   IF C="Z" THEN GO TO 76
76   IF C=" " THEN GO TO 77
77   IF C="." THEN GO TO 78
78   IF C="-" THEN GO TO 79
79   IF C=" " THEN GO TO 80
80   IF C="X" THEN GO TO 81
81   IF C="Y" THEN GO TO 82
82   IF C="Z" THEN GO TO 83
83   IF C="A" THEN GO TO 84
84   IF C="B" THEN GO TO 85
85   IF C="C" THEN GO TO 86
86   IF C="D" THEN GO TO 87
87   IF C="E" THEN GO TO 88
88   IF C="F" THEN GO TO 89
89   IF C="G" THEN GO TO 90
90   IF C="H" THEN GO TO 91
91   IF C="I" THEN GO TO 92
92   IF C="J" THEN GO TO 93
93   IF C="K" THEN GO TO 94
94   IF C="L" THEN GO TO 95
95   IF C="M" THEN GO TO 96
96   IF C="N" THEN GO TO 97
97   IF C="O" THEN GO TO 98
98   IF C="P" THEN GO TO 99
99   IF C="Q" THEN GO TO 100
100  IF C="R" THEN GO TO 101
101  IF C="S" THEN GO TO 102
102  IF C="T" THEN GO TO 103
103  IF C="U" THEN GO TO 104
104  IF C="V" THEN GO TO 105
105  IF C="W" THEN GO TO 106
106  IF C="X" THEN GO TO 107
107  IF C="Y" THEN GO TO 108
108  IF C="Z" THEN GO TO 109
109  IF C=" " THEN GO TO 110
110  IF C="." THEN GO TO 111
111  IF C="-" THEN GO TO 112
112  IF C=" " THEN GO TO 113
113  IF C="X" THEN GO TO 114
114  IF C="Y" THEN GO TO 115
115  IF C="Z" THEN GO TO 116
116  IF C="A" THEN GO TO 117
117  IF C="B" THEN GO TO 118
118  IF C="C" THEN GO TO 119
119  IF C="D" THEN GO TO 120
120  IF C="E" THEN GO TO 121
121  IF C="F" THEN GO TO 122
122  IF C="G" THEN GO TO 123
123  IF C="H" THEN GO TO 124
124  IF C="I" THEN GO TO 125
125  IF C="J" THEN GO TO 126
126  IF C="K" THEN GO TO 127
127  IF C="L" THEN GO TO 128
128  IF C="M" THEN GO TO 129
129  IF C="N" THEN GO TO 130
130  IF C="O" THEN GO TO 131
131  IF C="P" THEN GO TO 132
132  IF C="Q" THEN GO TO 133
133  IF C="R" THEN GO TO 134
134  IF C="S" THEN GO TO 135
135  IF C="T" THEN GO TO 136
136  IF C="U" THEN GO TO 137
137  IF C="V" THEN GO TO 138
138  IF C="W" THEN GO TO 139
139  IF C="X" THEN GO TO 140
140  IF C="Y" THEN GO TO 141
141  IF C="Z" THEN GO TO 142
142  IF C=" " THEN GO TO 143
143  IF C="." THEN GO TO 144
144  IF C="-" THEN GO TO 145
145  IF C=" " THEN GO TO 146
146  IF C="X" THEN GO TO 147
147  IF C="Y" THEN GO TO 148
148  IF C="Z" THEN GO TO 149
149  IF C="A" THEN GO TO 150
150  IF C="B" THEN GO TO 151
151  IF C="C" THEN GO TO 152
152  IF C="D" THEN GO TO 153
153  IF C="E" THEN GO TO 154
154  IF C="F" THEN GO TO 155
155  IF C="G" THEN GO TO 156
156  IF C="H" THEN GO TO 157
157  IF C="I" THEN GO TO 158
158  IF C="J" THEN GO TO 159
159  IF C="K" THEN GO TO 160
160  IF C="L" THEN GO TO 161
161  IF C="M" THEN GO TO 162
162  IF C="N" THEN GO TO 163
163  IF C="O" THEN GO TO 164
164  IF C="P" THEN GO TO 165
165  IF C="Q" THEN GO TO 166
166  IF C="R" THEN GO TO 167
167  IF C="S" THEN GO TO 168
168  IF C="T" THEN GO TO 169
169  IF C="U" THEN GO TO 170
170  IF C="V" THEN GO TO 171
171  IF C="W" THEN GO TO 172
172  IF C="X" THEN GO TO 173
173  IF C="Y" THEN GO TO 174
174  IF C="Z" THEN GO TO 175
175  IF C=" " THEN GO TO 176
176  IF C="." THEN GO TO 177
177  IF C="-" THEN GO TO 178
178  IF C=" " THEN GO TO 179
179  IF C="X" THEN GO TO 180
180  IF C="Y" THEN GO TO 181
181  IF C="Z" THEN GO TO 182
182  IF C="A" THEN GO TO 183
183  IF C="B" THEN GO TO 184
184  IF C="C" THEN GO TO 185
185  IF C="D" THEN GO TO 186
186  IF C="E" THEN GO TO 187
187  IF C="F" THEN GO TO 188
188  IF C="G" THEN GO TO 189
189  IF C="H" THEN GO TO 190
190  IF C="I" THEN GO TO 191
191  IF C="J" THEN GO TO 192
192  IF C="K" THEN GO TO 193
193  IF C="L" THEN GO TO 194
194  IF C="M" THEN GO TO 195
195  IF C="N" THEN GO TO 196
196  IF C="O" THEN GO TO 197
197  IF C="P" THEN GO TO 198
198  IF C="Q" THEN GO TO 199
199  IF C="R" THEN GO TO 200
200  IF C="S" THEN GO TO 201
201  IF C="T" THEN GO TO 202
202  IF C="U" THEN GO TO 203
203  IF C="V" THEN GO TO 204
204  IF C="W" THEN GO TO 205
205  IF C="X" THEN GO TO 206
206  IF C="Y" THEN GO TO 207
207  IF C="Z" THEN GO TO 208
208  IF C=" " THEN GO TO 209
209  IF C="." THEN GO TO 210
210  IF C="-" THEN GO TO 211
211  IF C=" " THEN GO TO 212
212  IF C="X" THEN GO TO 213
213  IF C="Y" THEN GO TO 214
214  IF C="Z" THEN GO TO 215
215  IF C="A" THEN GO TO 216
216  IF C="B" THEN GO TO 217
217  IF C="C" THEN GO TO 218
218  IF C="D" THEN GO TO 219
219  IF C="E" THEN GO TO 220
220  IF C="F" THEN GO TO 221
221  IF C="G" THEN GO TO 222
222  IF C="H" THEN GO TO 223
223  IF C="I" THEN GO TO 224
224  IF C="J" THEN GO TO 225
225  IF C="K" THEN GO TO 226
226  IF C="L" THEN GO TO 227
227  IF C="M" THEN GO TO 228
228  IF C="N" THEN GO TO 229
229  IF C="O" THEN GO TO 230
230  IF C="P" THEN GO TO 231
231  IF C="Q" THEN GO TO 232
232  IF C="R" THEN GO TO 233
233  IF C="S" THEN GO TO 234
234  IF C="T" THEN GO TO 235
235  IF C="U" THEN GO TO 236
236  IF C="V" THEN GO TO 237
237  IF C="W" THEN GO TO 238
238  IF C="X" THEN GO TO 239
239  IF C="Y" THEN GO TO 240
240  IF C="Z" THEN GO TO 241
241  IF C=" " THEN GO TO 242
242  IF C="." THEN GO TO 243
243  IF C="-" THEN GO TO 244
244  IF C=" " THEN GO TO 245
245  IF C="X" THEN GO TO 246
246  IF C="Y" THEN GO TO 247
247  IF C="Z" THEN GO TO 248
248  IF C="A" THEN GO TO 249
249  IF C="B" THEN GO TO 250
250  IF C="C" THEN GO TO 251
251  IF C="D" THEN GO TO 252
252  IF C="E" THEN GO TO 253
253  IF C="F" THEN GO TO 254
254  IF C="G" THEN GO TO 255
255  IF C="H" THEN GO TO 256
256  IF C="I" THEN GO TO 257
257  IF C="J" THEN GO TO 258
258  IF C="K" THEN GO TO 259
259  IF C="L" THEN GO TO 260
260  IF C="M" THEN GO TO 261
261  IF C="N" THEN GO TO 262
262  IF C="O" THEN GO TO 263
263  IF C="P" THEN GO TO 264
264  IF C="Q" THEN GO TO 265
265  IF C="R" THEN GO TO 266
266  IF C="S" THEN GO TO 267
267  IF C="T" THEN GO TO 268
268  IF C="U" THEN GO TO 269
269  IF C="V" THEN GO TO 270
270  IF C="W" THEN GO TO 271
271  IF C="X" THEN GO TO 272
272  IF C="Y" THEN GO TO 273
273  IF C="Z" THEN GO TO 274
274  IF C=" " THEN GO TO 275
275  IF C="." THEN GO TO 276
276  IF C="-" THEN GO TO 277
277  IF C=" " THEN GO TO 278
278  IF C="X" THEN GO TO 279
279  IF C="Y" THEN GO TO 280
280  IF C="Z" THEN GO TO 281
281  IF C="A" THEN GO TO 282
282  IF C="B" THEN GO TO 283
283  IF C="C" THEN GO TO 284
284  IF C="D" THEN GO TO 285
285  IF C="E" THEN GO TO 286
286  IF C="F" THEN GO TO 287
287  IF C="G" THEN GO TO 288
288  IF C="H" THEN GO TO 289
289  IF C="I" THEN GO TO 290
290  IF C="J" THEN GO TO 291
291  IF C="K" THEN GO TO 292
292  IF C="L" THEN GO TO 293
293  IF C="M" THEN GO TO 294
294  IF C="N" THEN GO TO 295
295  IF C="O" THEN GO TO 296
296  IF C="P" THEN GO TO 297
297  IF C="Q" THEN GO TO 298
298  IF C="R" THEN GO TO 299
299  IF C="S" THEN GO TO 300
300  IF C="T" THEN GO TO 301
301  IF C="U" THEN GO TO 302
302  IF C="V" THEN GO TO 303
303  IF C="W" THEN GO TO 304
304  IF C="X" THEN GO TO 305
305  IF C="Y" THEN GO TO 306
306  IF C="Z" THEN GO TO 307
307  IF C=" " THEN GO TO 308
308  IF C="." THEN GO TO 309
309  IF C="-" THEN GO TO 310
310  IF C=" " THEN GO TO 311
311  IF C="X" THEN GO TO 312
312  IF C="Y" THEN GO TO 313
313  IF C="Z" THEN GO TO 314
314  IF C="A" THEN GO TO 315
315  IF C="B" THEN GO TO 316
316  IF C="C" THEN GO TO 317
317  IF C="D" THEN GO TO 318
318  IF C="E" THEN GO TO 319
319  IF C="F" THEN GO TO 320
320  IF C="G" THEN GO TO 321
321  IF C="H" THEN GO TO 322
322  IF C="I" THEN GO TO 323
323  IF C="J" THEN GO TO 324
324  IF C="K" THEN GO TO 325
325  IF C="L" THEN GO TO 326
326  IF C="M" THEN GO TO 327
327  IF C="N" THEN GO TO 328
328  IF C="O" THEN GO TO 329
329  IF C="P" THEN GO TO 330
330  IF C="Q" THEN GO TO 331
331  IF C="R" THEN GO TO 332
332  IF C="S" THEN GO TO 333
333  IF C="T" THEN GO TO 334
334  IF C="U" THEN GO TO 335
335  IF C="V" THEN GO TO 336
336  IF C="W" THEN GO TO 337
337  IF C="X" THEN GO TO 338
338  IF C="Y" THEN GO TO 339
339  IF C="Z" THEN GO TO 340
340  IF C=" " THEN GO TO 341
341  IF C="." THEN GO TO 342
342  IF C="-" THEN GO TO 343
343  IF C=" " THEN GO TO 344
344  IF C="X" THEN GO TO 345
345  IF C="Y" THEN GO TO 346
346  IF C="Z" THEN GO TO 347
347  IF C="A" THEN GO TO 348
348  IF C="B" THEN GO TO 349
349  IF C="C" THEN GO TO 350
350  IF C="D" THEN GO TO 351
351  IF C="E" THEN GO TO 352
352  IF C="F" THEN GO TO 353
353  IF C="G" THEN GO TO 354
354  IF C="H" THEN GO TO 355
355  IF C="I" THEN GO TO 356
356  IF C="J" THEN GO TO 357
357  IF C="K" THEN GO TO 358
358  IF C="L" THEN GO TO 359
359  IF C="M" THEN GO TO 360
360  IF C="N" THEN GO TO 361
361  IF C="O" THEN GO TO 362
362  IF C="P" THEN GO TO 363
363  IF C="Q" THEN GO TO 364
364  IF C="R" THEN GO TO 365
365  IF C="S" THEN GO TO 366
366  IF C="T" THEN GO TO 367
367  IF C="U" THEN GO TO 368
368  IF C="V" THEN GO TO 369
369  IF C="W" THEN GO TO 370
370  IF C="X" THEN GO TO 371
371  IF C="Y" THEN GO TO 372
372  IF C="Z" THEN GO TO 373
373  IF C=" " THEN GO TO 374
374  IF C="." THEN GO TO 375
375  IF C="-" THEN GO TO 376
376  IF C=" " THEN GO TO 377
377  IF C="X" THEN GO TO 378
378  IF C="Y" THEN GO TO 379
379  IF C="Z" THEN GO TO 380
380  IF C="A" THEN GO TO 381
381  IF C="B" THEN GO TO 382
382  IF C="C" THEN GO TO 383
383  IF C="D" THEN GO TO 384
384  IF C="E" THEN GO TO 385
385  IF C="F" THEN GO TO 386
386  IF C="G" THEN GO TO 387
387  IF C="H" THEN GO TO 388
388  IF C="I" THEN GO TO 389
389  IF C="J" THEN GO TO 390
390  IF C="K" THEN GO TO 391
391  IF C="L" THEN GO TO 392
392  IF C="M" THEN GO TO 393
393  IF C="N" THEN GO TO 394
394  IF C="O" THEN GO TO 395
395  IF C="P" THEN GO TO 396
396  IF C="Q" THEN GO TO 397
397  IF C="R" THEN GO TO 398
398  IF C="S" THEN GO TO 399
399  IF C="T" THEN GO TO 400
400  IF C="U" THEN GO TO 401
401  IF C="V" THEN GO TO 402
402  IF C="W" THEN GO TO 403
403  IF C="X" THEN GO TO 404
404  IF C="Y" THEN GO TO 405
405  IF C="Z" THEN GO TO 406
406  IF C=" " THEN GO TO 407
407  IF C="." THEN GO TO 408
408  IF C="-" THEN GO TO 409
409  IF C=" " THEN GO TO 410
410  IF C="X" THEN GO TO 411
411  IF C="Y" THEN GO TO 412
412  IF C="Z" THEN GO TO 413
413  IF C="A" THEN GO TO 414
414  IF C="B" THEN GO TO 415
415  IF C="C" THEN GO TO 416
416  IF C="D" THEN GO TO 417
417  IF C="E" THEN GO TO 418
418  IF C="F" THEN GO TO 419
419  IF C="G" THEN GO TO 420
420  IF C="H" THEN GO TO 421
421  IF C="I" THEN GO TO 422
422  IF C="J" THEN GO TO 423
423  IF C="K" THEN GO TO 424
424  IF C="L" THEN GO TO 425
425  IF C="M" THEN GO TO 426
426  IF C="N" THEN GO TO 427
427  IF C="O" THEN GO TO 428
428  IF C="P" THEN GO TO 429
429  IF C="Q" THEN GO TO 430
430  IF C="R" THEN GO TO 431
431  IF C="S" THEN GO TO 432
432  IF C="T" THEN GO TO 433
433  IF C="U" THEN GO TO 434
434  IF C="V" THEN GO TO 435
435  IF C="W" THEN GO TO 436
436  IF C="X" THEN GO TO 437
437  IF C="Y" THEN GO TO 438
438  IF C="Z" THEN GO TO 439
439  IF C=" " THEN GO TO 440
440  IF C="." THEN GO TO 441
441  IF C="-" THEN GO TO 442
442  IF C=" " THEN GO TO 443
443  IF C="X" THEN GO TO 444
444  IF C="Y" THEN GO TO 445
445  IF C="Z" THEN GO TO 446
446  IF C="A" THEN GO TO 447
447  IF C="B" THEN GO TO 448
448  IF C="C" THEN GO TO 449
449  IF C="D" THEN GO TO 450
450  IF C="E" THEN GO TO 451
451  IF C="F" THEN GO TO 452
452  IF C="G" THEN GO TO 453
453  IF C="H" THEN GO TO 454
454  IF C="I" THEN GO TO 455
455  IF C="J" THEN GO TO 456
456  IF C="K" THEN GO TO 457
457  IF C="L" THEN GO TO 458
458  IF C="M" THEN GO TO 459
459  IF C="N" THEN GO TO 460
460  IF C="O" THEN GO TO 461
461  IF C="P" THEN GO TO 462
462  IF C="Q" THEN GO TO 463
463  IF C="R" THEN GO TO 464
464  IF C="S" THEN GO TO 465
465  IF C="T" THEN GO TO 466
466  IF C="U" THEN GO TO 467
467  IF C="V" THEN GO TO 468
468  IF C="W" THEN GO TO 469
469  IF C="X" THEN GO TO 470
470  IF C="Y" THEN GO TO 471
471  IF C="Z" THEN GO TO 472
472  IF C=" " THEN GO TO 473
473  IF C="." THEN GO TO 474
474  IF C="-" THEN GO TO 475
475  IF C=" " THEN GO TO 476
476  IF C="X" THEN GO TO 477
477  IF C="Y" THEN GO TO 478
478  IF C="Z" THEN GO TO 479
479  IF C="A" THEN GO TO 480
480  IF C="B" THEN GO TO 481
481  IF C="C" THEN GO TO 482
482  IF C="D" THEN GO TO 483
483  IF C="E" THEN GO TO 484
484  IF C="F" THEN GO TO 485
485  IF C="G" THEN GO TO 486
486  IF C="H" THEN GO TO 487
487  IF C="I" THEN GO TO 488
488  IF C="J" THEN GO TO 489
489  IF C="K" THEN GO TO 490
490  IF C="L" THEN GO TO 491
491  IF C="M" THEN GO TO 492
492  IF C="N" THEN GO TO 493
493  IF C="O" THEN GO TO 494
494  IF C="P" THEN GO TO 495
495  IF C="Q" THEN GO TO 496
496  IF C="R" THEN GO TO 497
497  IF C="S" THEN GO TO 498
498  IF C="T" THEN GO TO 499
499  IF C="U" THEN GO TO 500
500  IF C="V" THEN GO TO 501
501  IF C="W" THEN GO TO 502
502  IF C="X" THEN GO TO 503
503  IF C="Y" THEN GO TO 504
504  IF C="Z" THEN GO TO 505
505  IF C=" " THEN GO TO 506
506  IF C="." THEN GO TO 507
507  IF C="-" THEN GO TO 508
508  IF C=" " THEN GO TO 509
509  IF C="X" THEN GO TO 510
510  IF C="Y" THEN GO TO 511
511  IF C="Z" THEN GO TO 512
512  IF C="A" THEN GO TO 513
513  IF C="B" THEN GO TO 514
514  IF C="C" THEN GO TO 515
515  IF C="D" THEN GO TO 516
516  IF C="E" THEN GO TO 517
517  IF C="F" THEN GO TO 518
518  IF C="G" THEN GO TO 519
519  IF C="H" THEN GO TO 520
520  IF C="I" THEN GO TO 521
521  IF C="J" THEN GO TO 522
522  IF C="K" THEN GO TO 523
523  IF C="L" THEN GO TO 524
524  IF C="M" THEN GO TO 525
525  IF C="N" THEN GO TO 526
526  IF C="O" THEN GO TO 527
527  IF C="P" THEN GO TO 528
528  IF C="Q" THEN GO TO 529
529  IF C="R" THEN GO TO 530
530  IF C="S" THEN GO TO 531
531  IF C="T" THEN GO TO 532
532  IF C="U" THEN GO TO 533
533  IF C="V" THEN GO TO 534
534  IF C="W" THEN GO TO 535
535  IF C="X" THEN GO TO 536
536  IF C="Y" THEN GO TO 537
537  IF C="Z" THEN GO TO 538
538  IF C=" " THEN GO TO 539
539  IF C="." THEN GO TO 540
540  IF C="-" THEN GO TO 541
541  IF C=" " THEN GO TO 542
542  IF C="X" THEN GO TO 543
543  IF C="Y" THEN GO TO 544
544  IF C="Z" THEN GO TO 545
545  IF C="A" THEN GO TO 546
546  IF C="B" THEN GO TO 547
547  IF C="C" THEN GO TO 548
548  IF C="D" THEN GO TO 549
549  IF C="E" THEN GO TO 550
550  IF C="F" THEN GO TO 551
551  IF C="G" THEN GO TO 552
552  IF C="H" THEN GO TO 553
553  IF C="I" THEN GO TO 554
554  IF C="J" THEN GO TO 555
555  IF C="K" THEN GO TO 556
556  IF C="L" THEN GO TO 557
557  IF C="M" THEN GO TO 558
558  IF C="N" THEN GO TO 559
559  IF C="O" THEN GO TO 560
560  IF C="P" THEN GO TO 561
561  IF C="Q" THEN GO TO 562
562  IF C="R" THEN GO TO 563
563  IF C="S" THEN GO TO 564
564  IF C="T" THEN GO TO 565
565  IF C="U" THEN GO TO 566
566  IF C="V" THEN GO TO 567
567  IF C="W" THEN GO TO 568
568  IF C="X" THEN GO TO 569
569  IF C="Y" THEN GO TO 570
570  IF C="Z" THEN GO TO 571
571  IF C=" " THEN GO TO 572
572  IF C="." THEN GO TO 573
573  IF C="-" THEN GO TO 574
574  IF C=" " THEN GO TO 575
575  IF C="X" THEN GO TO 576
576  IF C="Y" THEN GO TO 577
577  IF C="Z" THEN GO TO 578
578  IF C="A" THEN GO TO 579
579  IF C="B" THEN GO TO 580
580  IF C="C" THEN GO TO 581
581  IF C="D" THEN GO TO 582
582  IF C="E" THEN GO TO 583
583  IF C="F" THEN GO TO 584
584  IF C="G" THEN GO TO 585
585  IF C="H" THEN GO TO 586
586  IF C="I" THEN GO TO 587
587  IF C="J" THEN GO TO 588
588  IF C="K" THEN GO TO 589
589  IF C="L" THEN GO TO 590
590  IF C="M" THEN GO TO 591
591  IF C="N" THEN GO TO 592
592  IF C="O" THEN GO TO 593
593  IF C="P" THEN GO TO 594
594  IF C="Q" THEN GO TO 595
595  IF C="R" THEN GO TO 596
596  IF C="S" THEN GO TO 597
597  IF C="T" THEN GO TO 598
598  IF C="U" THEN GO TO 599
599  IF C="V" THEN GO TO 600
600  IF C="W" THEN GO TO 601
601  IF C="X" THEN GO TO 602
602  IF C="Y" THEN GO TO 603
603  IF C="Z" THEN GO TO 604
604  IF C=" " THEN GO TO 605
605  IF C="." THEN GO TO 606
606  IF C="-" THEN GO TO 607
607  IF C=" " THEN GO TO 608
608  IF C="X" THEN GO TO 609
609  IF C="Y" THEN GO TO 610
610  IF C="Z" THEN GO TO 611
611  IF C="A" THEN GO TO 612
612  IF C="B" THEN GO TO 613
613  IF C="C" THEN GO TO 614
614  IF C="D" THEN GO TO 615
615  IF C="E" THEN GO TO 616
616  IF C="F" THEN GO TO 617
617  IF C="G" THEN GO TO 618
618  IF C="H" THEN GO TO 619
619  IF C="I" THEN GO TO 620
620  IF C="J" THEN GO TO 621
621  IF C="K" THEN GO TO 622
622  IF C="L" THEN GO TO 623
623  IF C="M" THEN GO TO 624
624  IF C="N" THEN GO TO 625
625  IF C="O" THEN GO TO 626
626  IF C="P" THEN GO TO 627
627  IF C="Q" THEN GO TO 628
628  IF C="R" THEN GO TO 629
629  IF C="S" THEN GO TO 630
630  IF C="T" THEN GO TO 631
631  IF C="U" THEN GO TO 632
632  IF C="V" THEN GO TO 633
633  IF C="W" THEN GO TO 634
634  IF C="X" THEN GO TO 635
635  IF C="Y" THEN GO TO 636
636  IF C="Z" THEN GO TO 637
637  IF C=" " THEN GO TO 638
638  IF C="." THEN GO TO 639
639  IF C="-" THEN GO TO 640
640  IF C=" " THEN GO TO 641
641  IF C="X" THEN GO TO 642
642  IF C="Y" THEN GO TO 643
643  IF C="Z" THEN GO TO 644
644  IF C="A" THEN GO TO 645
645  IF C="B" THEN GO TO 646
646  IF C="C" THEN GO TO 647
647  IF C="D" THEN GO TO 648
648  IF C="E" THEN GO TO 649
649  IF C="F" THEN GO TO 650
650  IF C="G" THEN GO TO 651
651  IF C="H" THEN GO TO 652
652  IF C="I" THEN GO TO 653
653  IF C="J" THEN GO TO 654
654  IF C="K" THEN GO TO 655
655  IF C="L" THEN GO TO 656
656  IF C="M" THEN GO TO 657
657  IF C="N" THEN GO TO 658
658  IF C="O" THEN GO TO 659
659  IF C="P" THEN GO TO 660
660  IF C="Q" THEN GO TO 661
661  IF C="R" THEN GO TO 662
662  IF C="S" THEN GO TO 663
663  IF C="T" THEN GO TO 664
664  IF C="U" THEN GO TO 665
665  IF C="V" THEN GO TO 666
666  IF C="W" THEN GO TO 667
667  IF C="X" THEN GO TO 668
668  IF C="Y" THEN GO TO 669
669  IF C="Z" THEN GO TO 670
670  IF C=" " THEN GO TO 671
671  IF C="." THEN GO TO 672
672  IF C="-" THEN GO TO 673
673  IF C=" " THEN GO TO 674
674  IF C="X" THEN GO TO 675
675  IF C="Y" THEN GO TO 676
676  IF C="Z" THEN GO TO 677
677  IF C="A" THEN GO TO 678
678  IF C="B" THEN GO TO 679
679  IF C="C" THEN GO TO 680
680  IF C="D" THEN GO TO 681
681  IF C="E" THEN GO TO 682
682  IF C="F" THEN GO TO 683
683  IF C="G" THEN GO TO 684
684  IF C="H" THEN GO TO 685
685  IF C="I" THEN GO TO 686
686  IF C="J" THEN GO TO 687
687  IF C="K" THEN GO TO 688
688  IF C="L
```

R&R**SOFTWARE**

means value



At R&R we believe that value comes first, and we make it our business to ensure that we provide good quality programs at realistic prices... for your enjoyment.

ALL ARCADE GAMES NORMALLY HAVE CHOICE OF KEYBOARD OR JOYSTICK CONTROL.

Should you have difficulty in obtaining our products from your local Dealer please send cheque or Postal Order indicating if fee required, for return of post service.

CHOPPER X-1

For any 16 Spectrum

Two dimensional Spectrum version of a famous... helicopter to make manoeuvring and take the air... an action packed original game that's highly fun.

... £5.50

GALAXY WARLORDS

For any 16 Spectrum

A fast paced arcade style space game in which... players manoeuvre their ships around... showing enemy movement. With your skills this... dimensional version. The only new version... that can be played on Spectrum and Amstrad.

... £5.50

JUNIOR ARITHMETIC

For any 16 Spectrum

A simple mathematical program to teach the... for any major computer system. Includes... Spectrum, Amstrad, BBC, and Acorn. It... can be used as a teaching aid or as a... for any major computer system.

... £4.95

SPECTIPEDE

For any 16 Spectrum

A simple but addictive game which uses your... to make the most of the Spectrum's... and Amstrad. It's a fast paced... and Amstrad. It's a fast paced...

... £5.50

GOLF

For any 16 Spectrum

Enjoy a game of golf with your... with a choice of 16... and Amstrad. It's a fast paced... and Amstrad. It's a fast paced...

... £3.75

GNASHER

For any 16 Spectrum

A fast paced arcade style game with... and Amstrad. It's a fast paced... and Amstrad. It's a fast paced...

... £4.95

STAR TREK

For any 16 Spectrum

A fast paced arcade style game with... and Amstrad. It's a fast paced... and Amstrad. It's a fast paced...

... £4.95

OTHER COMPUTERS?

Send us a list of your computer... and Amstrad. It's a fast paced... and Amstrad. It's a fast paced...

R&R**SOFTWARE**

DEALER ENQUIRIES WELCOME

R&R Software Ltd.

6 Russell Street, Gloucester GL1 1NE.

Tel (0452) 502813

CALLING ALL PROGRAMMERS...

WANTED: New, Quality Software.
Send us your latest Programs
for evaluation - NOW.

AVAILABLE FROM LEADING DEALERS NATIONWIDE

ALIENS

Can you defend the Earth from the invading aliens?

In this game for your IGG Spectrum, you command a missile base which you can maneuver along the surface of the Earth, using the 'p' key to move right and the 'o' key to move left. To fire a missile you have to press the 'q' key.

The aliens, in their usual

unfriendly way, re-emerge at different levels above the Earth one at a time. It is your job to shoot them out of the sky before they get enough ships out of hyperspace to destroy the Earth. Once they have 10 ships positioned over the surface, they will be in a position to do

this. So, your mission, should you choose to accept it, is to keep the number of aliens in the sky down to less than 10.

You can only fire one missile at a time on-screen, but if you find it too difficult to play at one level, you can always choose another. *A*



C.B. Spectrum



```

10 GO TO 1000
20 GO TO 1000
30 REM ***** SETUP
40 BORDER 0
45 CLR : PRINT AT 20,2: PAPER
50 PRINT AT 21,2: PAPER 1
60 FOR j=1 TO 20
70 CLY: AND=204: AND=120+40
80 NEXT j
90 LET i=0: LET l=10
100 REM *****
110 REM *****
120 REM *****
130 IF 1-AND=120 THEN POINT 0
140 IF 1-AND=204 THEN INK 2: "0": LE
150 IF 1-AND=120 THEN GO TO 400
160 IF 1-AND=204 THEN GO TO 400
170 IF 1-AND=120 THEN GO TO 400
180 IF 1-AND=204 THEN GO TO 400
190 IF 1-AND=120 THEN GO TO 400
200 IF 1-AND=204 THEN GO TO 400
210 IF 1-AND=120 THEN GO TO 400
220 IF 1-AND=204 THEN GO TO 400
230 IF 1-AND=120 THEN GO TO 400
240 IF 1-AND=204 THEN GO TO 400
250 IF 1-AND=120 THEN GO TO 400
260 IF 1-AND=204 THEN GO TO 400
270 IF 1-AND=120 THEN GO TO 400
280 IF 1-AND=204 THEN GO TO 400
290 IF 1-AND=120 THEN GO TO 400
300 IF 1-AND=204 THEN GO TO 400
310 IF 1-AND=120 THEN GO TO 400
320 IF 1-AND=204 THEN GO TO 400
330 IF 1-AND=120 THEN GO TO 400
340 IF 1-AND=204 THEN GO TO 400
350 IF 1-AND=120 THEN GO TO 400
360 IF 1-AND=204 THEN GO TO 400
370 IF 1-AND=120 THEN GO TO 400
380 IF 1-AND=204 THEN GO TO 400
390 IF 1-AND=120 THEN GO TO 400
400 IF 1-AND=204 THEN GO TO 400
410 IF 1-AND=120 THEN GO TO 400
420 IF 1-AND=204 THEN GO TO 400
430 IF 1-AND=120 THEN GO TO 400
440 IF 1-AND=204 THEN GO TO 400
450 IF 1-AND=120 THEN GO TO 400
460 IF 1-AND=204 THEN GO TO 400
470 IF 1-AND=120 THEN GO TO 400
480 IF 1-AND=204 THEN GO TO 400
490 IF 1-AND=120 THEN GO TO 400
500 IF 1-AND=204 THEN GO TO 400
510 IF 1-AND=120 THEN GO TO 400
520 IF 1-AND=204 THEN GO TO 400
530 IF 1-AND=120 THEN GO TO 400
540 IF 1-AND=204 THEN GO TO 400
550 IF 1-AND=120 THEN GO TO 400
560 IF 1-AND=204 THEN GO TO 400
570 IF 1-AND=120 THEN GO TO 400
580 IF 1-AND=204 THEN GO TO 400
590 IF 1-AND=120 THEN GO TO 400
600 IF 1-AND=204 THEN GO TO 400
610 IF 1-AND=120 THEN GO TO 400
620 IF 1-AND=204 THEN GO TO 400
630 IF 1-AND=120 THEN GO TO 400
640 IF 1-AND=204 THEN GO TO 400
650 IF 1-AND=120 THEN GO TO 400
660 IF 1-AND=204 THEN GO TO 400
670 IF 1-AND=120 THEN GO TO 400
680 IF 1-AND=204 THEN GO TO 400
690 IF 1-AND=120 THEN GO TO 400
700 IF 1-AND=204 THEN GO TO 400
710 IF 1-AND=120 THEN GO TO 400
720 IF 1-AND=204 THEN GO TO 400
730 IF 1-AND=120 THEN GO TO 400
740 IF 1-AND=204 THEN GO TO 400
750 IF 1-AND=120 THEN GO TO 400
760 IF 1-AND=204 THEN GO TO 400
770 IF 1-AND=120 THEN GO TO 400
780 IF 1-AND=204 THEN GO TO 400
790 IF 1-AND=120 THEN GO TO 400
800 IF 1-AND=204 THEN GO TO 400
810 IF 1-AND=120 THEN GO TO 400
820 IF 1-AND=204 THEN GO TO 400
830 IF 1-AND=120 THEN GO TO 400
840 IF 1-AND=204 THEN GO TO 400
850 IF 1-AND=120 THEN GO TO 400
860 IF 1-AND=204 THEN GO TO 400
870 IF 1-AND=120 THEN GO TO 400
880 IF 1-AND=204 THEN GO TO 400
890 IF 1-AND=120 THEN GO TO 400
900 IF 1-AND=204 THEN GO TO 400
910 IF 1-AND=120 THEN GO TO 400
920 IF 1-AND=204 THEN GO TO 400
930 IF 1-AND=120 THEN GO TO 400
940 IF 1-AND=204 THEN GO TO 400
950 IF 1-AND=120 THEN GO TO 400
960 IF 1-AND=204 THEN GO TO 400
970 IF 1-AND=120 THEN GO TO 400
980 IF 1-AND=204 THEN GO TO 400
990 IF 1-AND=120 THEN GO TO 400
1000 RETURN

```

LEPRECHAUN'S GOLD

An amazing game for your 16K Spectrum.



This 3D maze game for the Spectrum does not have dinosaurs charging up on you but it does have a Green Goblin. He follows you about, sending an indecisive player back to the start, and giving you a pot of gold if you get to the centre.

INTERESTING POINTS

1. No maze plan is shown — There are two reasons for this: a) I think it makes the game too easy and b) the computer does not have a clue about the shape of the maze until play begins — it all depends on the values of RND in line 260 and the moves a player makes. The maze does, however, once set, remain constant throughout the game (strings x3 and y3 assure that).
2. The program does not use machine code — I feel that the speed of the Spectrum's PLOT and LIN DRAW commands are fast enough. And, by avoiding machine code, the structure of the program is more easily seen by those who like to dissect programs.

3. The problem of 16K being about 961 — This has had to result in a compromise:
 - a) The elements of the maze (1000-1710) use, for

the most part, numbers and are placed at the beginning of the listing so that they are displayed quickly.

b) The parts which can operate slower are put later and use variables and 'VALs' to save bytes.
c) The UDGs are set separately and are SAVED onto tape as bytes to be LOADED in by the main program.

NOTES ON ENTERING THE PROGRAM

1. As explained above the UDGs have to be set first. Type in the program starting at line 1 to line 30 and RUN this. This will set the UDGs above RAMTOP. Now enter the rest of the program and SAVE it, followed by the UDG bytes using command:
- 2.

Table 1. A breakdown of the program.
Leprechaun's gold, line by line.

Lines	Description
190-280	: Logic controlling the patch through the maze (which is set up in line 1000 — see later). The basis is this: a) 210 — If the next-but-one element is a dead-end (7) then generate a random number. If this is less than 0.2 then the routine skips the dead-end, otherwise the dead-end is reached and the player is sent back to the beginning. b) 270 — When the player reaches this decision point again, if he or she chooses to take an alternate pathway (the original choice being held in string y3) then the dead end is avoided. c) 280 — If during a further attempt the player takes a turning different to the first attempt (excepting (b) above) then a dead end is reached and he or she is sent back to the beginning.
1000-1710	: The elements of the maze are displayed on the screen in 3D as it looking down the passage. There are three passage elements (1000: left turn, 1001: right turn and 1200: straight on) and four junction elements (1300: forward or right, 1400: forward or left, 1500: left or right and 1600: forward, left or

right) plus the dead-end at 1700. These are accessed in passage-junction pairs by line 200. Each element (except those at lines 1200 and 1700) are two part giving first a distant view followed by one with the player (see 4001) in a position to turn.

- 2000-2020 : If the player tries to walk into the passage wall this routine prevents him or her from doing so.
- 3000-3060 : This section invites a choice to be made (allowing only '5', '7' or '8'). If one is not made in about three seconds then the Green Goblin appears and sends the player back to the start.
- 4001 : This displays the player as a stick man, with his hands on his hips.
- 4600 : A complex logic routine which, on the / command '5' or '8', causes the figure to walk to the left or right and disappear around the corner of the passage.
- 4800 : This causes the figure to move forwards on the command '7'.
- 5000 : This turns the figure to the left or right before walking (or bumping into the wall).
- 6000-6035 : On LOADING, this is where the program begins. Instructions are given, variables are set, and the difficulty level (the number of elements in the maze) is chosen.
- 7000-7999 : The winning sequence. The gold is reached (much to the chagrin of the Green Goblin) and the player is invited to play again. It should be noted that the program can be stopped at any junction by pressing '0' in place of '5', '7' or '8' — see 3050.
- 8000 : Here the maze is set up as a string of numbers (representing the elements) held in 25. They are alternately 0-2 and 3-7, these not being two 7s following (a junction would have two dead-end exits).
- 9000 : This LOADs the user defined graphics (UDGs).

SAVE "maze" LINE 9000
SAVE "maze" CODE
USR4", 169

You will get the usual 'start tape then press any key' message. Do so, but don't walk away to make a cup of coffee! After about 45 seconds, the message will appear again. Don't stop the tape, just press any key and the UDGs bytes will be saved.

To verify use the command:

VERIFY "

After (hopefully) the 'OK' message appears stop the tape promptly, then verify the UDGs with:

VERIFY " CODE.

3. To LOAD simply enter LOAD "maze". The main program will LOAD, then RUN (and from line 9000 which will LOAD in the UDGs (following on tape) before displaying the instructions, and away you go.

Hope you find the GOLD!

As a final note, the single or pairs of capital letters shown in the LISTING within quotes are the UDGs and should be entered in the G mode.

```

1 FOR I=1 TO 16: READ A$: FOR
2  A=0 TO 7: READ A: POKE USR4 P&40
3  NEXT A: NEXT I
4 DATA "0",00,00,00,00,00,00,
5 0,00
6 DATA "1",00,00,00,00,00,00,0
7 0,00
8 DATA "2",00,00,00,00,00,00,0
9 0,00
10 DATA "3",00,00,00,00,00,00,0
11 0,00
12 DATA "4",00,00,00,00,00,00,0
13 0,00
14 DATA "5",00,00,00,00,00,00,0
15 0,00
16 DATA "6",00,00,00,00,00,00,0
17 DATA "7",00,00,00,00,00,00,0
18 DATA "8",00,00,00,00,00,00,0
19 DATA "9",00,00,00,00,00,00,0
20 DATA "A",00,00,00,00,00,00,0
21 DATA "B",00,00,00,00,00,00,0
22 DATA "C",00,00,00,00,00,00,0
23 DATA "D",00,00,00,00,00,00,0
24 DATA "E",00,00,00,00,00,00,0
25 DATA "F",00,00,00,00,00,00,0
26 DATA "G",00,00,00,00,00,00,0
27 DATA "H",00,00,00,00,00,00,0
28 DATA "I",00,00,00,00,00,00,0
29 DATA "J",00,00,00,00,00,00,0
30 DATA "K",00,00,00,00,00,00,0
31 DATA "L",00,00,00,00,00,00,0
32 DATA "M",00,00,00,00,00,00,0
33 DATA "N",00,00,00,00,00,00,0
34 DATA "O",00,00,00,00,00,00,0
35 DATA "P",00,00,00,00,00,00,0
36 DATA "Q",00,00,00,00,00,00,0
37 DATA "R",00,00,00,00,00,00,0
38 DATA "S",00,00,00,00,00,00,0
39 DATA "T",00,00,00,00,00,00,0
40 DATA "U",00,00,00,00,00,00,0
41 DATA "V",00,00,00,00,00,00,0
42 DATA "W",00,00,00,00,00,00,0
43 DATA "X",00,00,00,00,00,00,0
44 DATA "Y",00,00,00,00,00,00,0
45 DATA "Z",00,00,00,00,00,00,0
46 DATA "0",00,00,00,00,00,00,0
47 DATA "1",00,00,00,00,00,00,0
48 DATA "2",00,00,00,00,00,00,0
49 DATA "3",00,00,00,00,00,00,0
50 DATA "4",00,00,00,00,00,00,0
51 DATA "5",00,00,00,00,00,00,0
52 DATA "6",00,00,00,00,00,00,0
53 DATA "7",00,00,00,00,00,00,0
54 DATA "8",00,00,00,00,00,00,0
55 DATA "9",00,00,00,00,00,00,0
56 DATA "A",00,00,00,00,00,00,0
57 DATA "B",00,00,00,00,00,00,0
58 DATA "C",00,00,00,00,00,00,0
59 DATA "D",00,00,00,00,00,00,0
60 DATA "E",00,00,00,00,00,00,0
61 DATA "F",00,00,00,00,00,00,0
62 DATA "G",00,00,00,00,00,00,0
63 DATA "H",00,00,00,00,00,00,0
64 DATA "I",00,00,00,00,00,00,0
65 DATA "J",00,00,00,00,00,00,0
66 DATA "K",00,00,00,00,00,00,0
67 DATA "L",00,00,00,00,00,00,0
68 DATA "M",00,00,00,00,00,00,0
69 DATA "N",00,00,00,00,00,00,0
70 DATA "O",00,00,00,00,00,00,0
71 DATA "P",00,00,00,00,00,00,0
72 DATA "Q",00,00,00,00,00,00,0
73 DATA "R",00,00,00,00,00,00,0
74 DATA "S",00,00,00,00,00,00,0
75 DATA "T",00,00,00,00,00,00,0
76 DATA "U",00,00,00,00,00,00,0
77 DATA "V",00,00,00,00,00,00,0
78 DATA "W",00,00,00,00,00,00,0
79 DATA "X",00,00,00,00,00,00,0
80 DATA "Y",00,00,00,00,00,00,0
81 DATA "Z",00,00,00,00,00,00,0
82 DATA "0",00,00,00,00,00,00,0
83 DATA "1",00,00,00,00,00,00,0
84 DATA "2",00,00,00,00,00,00,0
85 DATA "3",00,00,00,00,00,00,0
86 DATA "4",00,00,00,00,00,00,0
87 DATA "5",00,00,00,00,00,00,0
88 DATA "6",00,00,00,00,00,00,0
89 DATA "7",00,00,00,00,00,00,0
90 DATA "8",00,00,00,00,00,00,0
91 DATA "9",00,00,00,00,00,00,0
92 DATA "A",00,00,00,00,00,00,0
93 DATA "B",00,00,00,00,00,00,0
94 DATA "C",00,00,00,00,00,00,0
95 DATA "D",00,00,00,00,00,00,0
96 DATA "E",00,00,00,00,00,00,0
97 DATA "F",00,00,00,00,00,00,0
98 DATA "G",00,00,00,00,00,00,0
99 DATA "H",00,00,00,00,00,00,0
100 DATA "I",00,00,00,00,00,00,0
101 DATA "J",00,00,00,00,00,00,0
102 DATA "K",00,00,00,00,00,00,0
103 DATA "L",00,00,00,00,00,00,0
104 DATA "M",00,00,00,00,00,00,0
105 DATA "N",00,00,00,00,00,00,0
106 DATA "O",00,00,00,00,00,00,0
107 DATA "P",00,00,00,00,00,00,0
108 DATA "Q",00,00,00,00,00,00,0
109 DATA "R",00,00,00,00,00,00,0
110 DATA "S",00,00,00,00,00,00,0
111 DATA "T",00,00,00,00,00,00,0
112 DATA "U",00,00,00,00,00,00,0
113 DATA "V",00,00,00,00,00,00,0
114 DATA "W",00,00,00,00,00,00,0
115 DATA "X",00,00,00,00,00,00,0
116 DATA "Y",00,00,00,00,00,00,0
117 DATA "Z",00,00,00,00,00,00,0
118 DATA "0",00,00,00,00,00,00,0
119 DATA "1",00,00,00,00,00,00,0
120 DATA "2",00,00,00,00,00,00,0
121 DATA "3",00,00,00,00,00,00,0
122 DATA "4",00,00,00,00,00,00,0
123 DATA "5",00,00,00,00,00,00,0
124 DATA "6",00,00,00,00,00,00,0
125 DATA "7",00,00,00,00,00,00,0
126 DATA "8",00,00,00,00,00,00,0
127 DATA "9",00,00,00,00,00,00,0
128 DATA "A",00,00,00,00,00,00,0
129 DATA "B",00,00,00,00,00,00,0
130 DATA "C",00,00,00,00,00,00,0
131 DATA "D",00,00,00,00,00,00,0
132 DATA "E",00,00,00,00,00,00,0
133 DATA "F",00,00,00,00,00,00,0
134 DATA "G",00,00,00,00,00,00,0
135 DATA "H",00,00,00,00,00,00,0
136 DATA "I",00,00,00,00,00,00,0
137 DATA "J",00,00,00,00,00,00,0
138 DATA "K",00,00,00,00,00,00,0
139 DATA "L",00,00,00,00,00,00,0
140 DATA "M",00,00,00,00,00,00,0
141 DATA "N",00,00,00,00,00,00,0
142 DATA "O",00,00,00,00,00,00,0
143 DATA "P",00,00,00,00,00,00,0
144 DATA "Q",00,00,00,00,00,00,0
145 DATA "R",00,00,00,00,00,00,0
146 DATA "S",00,00,00,00,00,00,0
147 DATA "T",00,00,00,00,00,00,0
148 DATA "U",00,00,00,00,00,00,0
149 DATA "V",00,00,00,00,00,00,0
150 DATA "W",00,00,00,00,00,00,0
151 DATA "X",00,00,00,00,00,00,0
152 DATA "Y",00,00,00,00,00,00,0
153 DATA "Z",00,00,00,00,00,00,0
154 DATA "0",00,00,00,00,00,00,0
155 DATA "1",00,00,00,00,00,00,0
156 DATA "2",00,00,00,00,00,00,0
157 DATA "3",00,00,00,00,00,00,0
158 DATA "4",00,00,00,00,00,00,0
159 DATA "5",00,00,00,00,00,00,0
160 DATA "6",00,00,00,00,00,00,0
161 DATA "7",00,00,00,00,00,00,0
162 DATA "8",00,00,00,00,00,00,0
163 DATA "9",00,00,00,00,00,00,0
164 DATA "A",00,00,00,00,00,00,0
165 DATA "B",00,00,00,00,00,00,0
166 DATA "C",00,00,00,00,00,00,0
167 DATA "D",00,00,00,00,00,00,0
168 DATA "E",00,00,00,00,00,00,0
169 DATA "F",00,00,00,00,00,00,0
170 DATA "G",00,00,00,00,00,00,0
171 DATA "H",00,00,00,00,00,00,0
172 DATA "I",00,00,00,00,00,00,0
173 DATA "J",00,00,00,00,00,00,0
174 DATA "K",00,00,00,00,00,00,0
175 DATA "L",00,00,00,00,00,00,0
176 DATA "M",00,00,00,00,00,00,0
177 DATA "N",00,00,00,00,00,00,0
178 DATA "O",00,00,00,00,00,00,0
179 DATA "P",00,00,00,00,00,00,0
180 DATA "Q",00,00,00,00,00,00,0
181 DATA "R",00,00,00,00,00,00,0
182 DATA "S",00,00,00,00,00,00,0
183 DATA "T",00,00,00,00,00,00,0
184 DATA "U",00,00,00,00,00,00,0
185 DATA "V",00,00,00,00,00,00,0
186 DATA "W",00,00,00,00,00,00,0
187 DATA "X",00,00,00,00,00,00,0
188 DATA "Y",00,00,00,00,00,00,0
189 DATA "Z",00,00,00,00,00,00,0
190 DATA "0",00,00,00,00,00,00,0
191 DATA "1",00,00,00,00,00,00,0
192 DATA "2",00,00,00,00,00,00,0
193 DATA "3",00,00,00,00,00,00,0
194 DATA "4",00,00,00,00,00,00,0
195 DATA "5",00,00,00,00,00,00,0
196 DATA "6",00,00,00,00,00,00,0
197 DATA "7",00,00,00,00,00,00,0
198 DATA "8",00,00,00,00,00,00,0
199 DATA "9",00,00,00,00,00,00,0
200 DATA "A",00,00,00,00,00,00,0
201 DATA "B",00,00,00,00,00,00,0
202 DATA "C",00,00,00,00,00,00,0
203 DATA "D",00,00,00,00,00,00,0
204 DATA "E",00,00,00,00,00,00,0
205 DATA "F",00,00,00,00,00,00,0
206 DATA "G",00,00,00,00,00,00,0
207 DATA "H",00,00,00,00,00,00,0
208 DATA "I",00,00,00,00,00,00,0
209 DATA "J",00,00,00,00,00,00,0
210 DATA "K",00,00,00,00,00,00,0
211 DATA "L",00,00,00,00,00,00,0
212 DATA "M",00,00,00,00,00,00,0
213 DATA "N",00,00,00,00,00,00,0
214 DATA "O",00,00,00,00,00,00,0
215 DATA "P",00,00,00,00,00,00,0
216 DATA "Q",00,00,00,00,00,00,0
217 DATA "R",00,00,00,00,00,00,0
218 DATA "S",00,00,00,00,00,00,0
219 DATA "T",00,00,00,00,00,00,0
220 DATA "U",00,00,00,00,00,00,0
221 DATA "V",00,00,00,00,00,00,0
222 DATA "W",00,00,00,00,00,00,0
223 DATA "X",00,00,00,00,00,00,0
224 DATA "Y",00,00,00,00,00,00,0
225 DATA "Z",00,00,00,00,00,00,0
226 DATA "0",00,00,00,00,00,00,0
227 DATA "1",00,00,00,00,00,00,0
228 DATA "2",00,00,00,00,00,00,0
229 DATA "3",00,00,00,00,00,00,0
230 DATA "4",00,00,00,00,00,00,0
231 DATA "5",00,00,00,00,00,00,0
232 DATA "6",00,00,00,00,00,00,0
233 DATA "7",00,00,00,00,00,00,0
234 DATA "8",00,00,00,00,00,00,0
235 DATA "9",00,00,00,00,00,00,0
236 DATA "A",00,00,00,00,00,00,0
237 DATA "B",00,00,00,00,00,00,0
238 DATA "C",00,00,00,00,00,00,0
239 DATA "D",00,00,00,00,00,00,0
240 DATA "E",00,00,00,00,00,00,0
241 DATA "F",00,00,00,00,00,00,0
242 DATA "G",00,00,00,00,00,00,0
243 DATA "H",00,00,00,00,00,00,0
244 DATA "I",00,00,00,00,00,00,0
245 DATA "J",00,00,00,00,00,00,0
246 DATA "K",00,00,00,00,00,00,0
247 DATA "L",00,00,00,00,00,00,0
248 DATA "M",00,00,00,00,00,00,0
249 DATA "N",00,00,00,00,00,00,0
250 DATA "O",00,00,00,00,00,00,0
251 DATA "P",00,00,00,00,00,00,0
252 DATA "Q",00,00,00,00,00,00,0
253 DATA "R",00,00,00,00,00,00,0
254 DATA "S",00,00,00,00,00,00,0
255 DATA "T",00,00,00,00,00,00,0
256 DATA "U",00,00,00,00,00,00,0
257 DATA "V",00,00,00,00,00,00,0
258 DATA "W",00,00,00,00,00,00,0
259 DATA "X",00,00,00,00,00,00,0
260 DATA "Y",00,00,00,00,00,00,0
261 DATA "Z",00,00,00,00,00,00,0
262 DATA "0",00,00,00,00,00,00,0
263 DATA "1",00,00,00,00,00,00,0
264 DATA "2",00,00,00,00,00,00,0
265 DATA "3",00,00,00,00,00,00,0
266 DATA "4",00,00,00,00,00,00,0
267 DATA "5",00,00,00,00,00,00,0
268 DATA "6",00,00,00,00,00,00,0
269 DATA "7",00,00,00,00,00,00,0
270 DATA "8",00,00,00,00,00,00,0
271 DATA "9",00,00,00,00,00,00,0
272 DATA "A",00,00,00,00,00,00,0
273 DATA "B",00,00,00,00,00,00,0
274 DATA "C",00,00,00,00,00,00,0
275 DATA "D",00,00,00,00,00,00,0
276 DATA "E",00,00,00,00,00,00,0
277 DATA "F",00,00,00,00,00,00,0
278 DATA "G",00,00,00,00,00,00,0
279 DATA "H",00,00,00,00,00,00,0
280 DATA "I",00,00,00,00,00,00,0
281 DATA "J",00,00,00,00,00,00,0
282 DATA "K",00,00,00,00,00,00,0
283 DATA "L",00,00,00,00,00,00,0
284 DATA "M",00,00,00,00,00,00,0
285 DATA "N",00,00,00,00,00,00,0
286 DATA "O",00,00,00,00,00,00,0
287 DATA "P",00,00,00,00,00,00,0
288 DATA "Q",00,00,00,00,00,00,0
289 DATA "R",00,00,00,00,00,00,0
290 DATA "S",00,00,00,00,00,00,0
291 DATA "T",00,00,00,00,00,00,0
292 DATA "U",00,00,00,00,00,00,0
293 DATA "V",00,00,00,00,00,00,0
294 DATA "W",00,00,00,00,00,00,0
295 DATA "X",00,00,00,00,00,00,0
296 DATA "Y",00,00,00,00,00,00,0
297 DATA "Z",00,00,00,00,00,00,0
298 DATA "0",00,00,00,00,00,00,0
299 DATA "1",00,00,00,00,00,00,0
300 DATA "2",00,00,00,00,00,00,0
301 DATA "3",00,00,00,00,00,00,0
302 DATA "4",00,00,00,00,00,00,0
303 DATA "5",00,00,00,00,00,00,0
304 DATA "6",00,00,00,00,00,00,0
305 DATA "7",00,00,00,00,00,00,0
306 DATA "8",00,00,00,00,00,00,0
307 DATA "9",00,00,00,00,00,00,0
308 DATA "A",00,00,00,00,00,00,0
309 DATA "B",00,00,00,00,00,00,0
310 DATA "C",00,00,00,00,00,00,0
311 DATA "D",00,00,00,00,00,00,0
312 DATA "E",00,00,00,00,00,00,0
313 DATA "F",00,00,00,00,00,00,0
314 DATA "G",00,00,00,00,00,00,0
315 DATA "H",00,00,00,00,00,00,0
316 DATA "I",00,00,00,00,00,00,0
317 DATA "J",00,00,00,00,00,00,0
318 DATA "K",00,00,00,00,00,00,0
319 DATA "L",00,00,00,00,00,00,0
320 DATA "M",00,00,00,00,00,00,0
321 DATA "N",00,00,00,00,00,00,0
322 DATA "O",00,00,00,00,00,00,0
323 DATA "P",00,00,00,00,00,00,0
324 DATA "Q",00,00,00,00,00,00,0
325 DATA "R",00,00,00,00,00,00,0
326 DATA "S",00,00,00,00,00,00,0
327 DATA "T",00,00,00,00,00,00,0
328 DATA "U",00,00,00,00,00,00,0
329 DATA "V",00,00,00,00,00,00,0
330 DATA "W",00,00,00,00,00,00,0
331 DATA "X",00,00,00,00,00,00,0
332 DATA "Y",00,00,00,00,00,00,0
333 DATA "Z",00,00,00,00,00,00,0
334 DATA "0",00,00,00,00,00,00,0
335 DATA "1",00,00,00,00,00,00,0
336 DATA "2",00,00,00,00,00,00,0
337 DATA "3",00,00,00,00,00,00,0
338 DATA "4",00,00,00,00,00,00,0
339 DATA "5",00,00,00,00,00,00,0
340 DATA "6",00,00,00,00,00,00,0
341 DATA "7",00,00,00,00,00,00,0
342 DATA "8",00,00,00,00,00,00,0
343 DATA "9",00,00,00,00,00,00,0
344 DATA "A",00,00,00,00,00,00,0
345 DATA "B",00,00,00,00,00,00,0
346 DATA "C",00,00,00,00,00,00,0
347 DATA "D",00,00,00,00,00,00,0
348 DATA "E",00,00,00,00,00,00,0
349 DATA "F",00,00,00,00,00,00,0
350 DATA "G",00,00,00,00,00,00,0
351 DATA "H",00,00,00,00,00,00,0
352 DATA "I",00,00,00,00,00,00,0
353 DATA "J",00,00,00,00,00,00,0
354 DATA "K",00,00,00,00,00,00,0
355 DATA "L",00,00,00,00,00,00,0
356 DATA "M",00,00,00,00,00,00,0
357 DATA "N",00,00,00,00,00,00,0
358 DATA "O",00,00,00,00,00,00,0
359 DATA "P",00,00,00,00,00,00,0
360 DATA "Q",00,00,00,00,00,00,0
361 DATA "R",00,00,00,00,00,00,0
362 DATA "S",00,00,00,00,00,00,0
363 DATA "T",00,00,00,00,00,00,0
364 DATA "U",00,00,00,00,00,00,0
365 DATA "V",00,00,00,00,00,00,0
366 DATA "W",00,00,00,00,00,00,0
367 DATA "X",00,00,00,00,00,00,0
368 DATA "Y",00,00,00,00,00,00,0
369 DATA "Z",00,00,00,00,00,00,0
370 DATA "0",00,00,00,00,00,00,0
371 DATA "1",00,00,00,00,00,00,0
372 DATA "2",00,00,00,00,00,00,0
373 DATA "3",00,00,00,00,00,00,0
374 DATA "4",00,00,00,00,00,00,0
375 DATA "5",00,00,00,00,00,00,0
376 DATA "6",00,00,00,00,00,00,0
377 DATA "7",00,00,00,00,00,00,0
378 DATA "8",00,00,00,00,00,00,0
379 DATA "9",00,00,00,00,00,00,0
380 DATA "A",00,00,00,00,00,00,0
381 DATA "B",00,00,00,00,00,00,0
382 DATA "C",00,00,00,00,00,00,0
383 DATA "D",00,00,00,00,00,00,0
384 DATA "E",00,00,00,00,00,00,0
385 DATA "F",00,00,00,00,00,00,0
386 DATA "G",00,00,00,00,00,00,0
387 DATA "H",00,00,00,00,00,00,0
388 DATA "I",00,00,00,00,00,00,0
389 DATA "J",00,00,00,00,00,00,0
390 DATA "K",00,00,00,00,00,00,0
391 DATA "L",00,00,00,00,00,00,0
392 DATA "M",00,00,00,00,00,00,0
393 DATA "N",00,00,00,00,00,00,0
394 DATA "O",00,00,00,00,00,00,0
395 DATA "P",00,00,00,00,00,00,0
396 DATA "Q",00,00,00,00,00,00,0
397 DATA "R",00,00,00,00,00,00,0
398 DATA "S",00,00,00,00,00,00,0
399 DATA "T",00,00,00,00,00,00,0
400 DATA "U",00,00,00,00,00,00,0
401 DATA "V",00,00,00,00,00,00,0
402 DATA "W",00,00,00,00,00,00,0
403 DATA "X",00,00,00,00,00,00,0
404 DATA "Y",00,00,00,00,00,00,0
405 DATA "Z",00,00,00,00,00,00,0
406 DATA "0",00,00,00,00,00,00,0
407 DATA "1",00,00,00,00,00,00,0
408 DATA "2",00,00,00,00,00,00,0
409 DATA "3",00,00,00,00,00,00,0
410 DATA "4",00,00,00,00,00,00,0
411 DATA "5",00,00,00,00,00,00,0
412 DATA "6",00,00,00,00,00,00,0
413 DATA "7",00,00,00,00,00,00,0
414 DATA "8",00,00,00,00,00,00,0
415 DATA "9",00,00,00,00,00,00,0
416 DATA "A",00,00,00,00,00,00,0
417 DATA "B",00,00,00,00,00,00,0
418 DATA "C",00,00,00,00,00,00,0
419 DATA "D",00,00,00,00,00,00,0
420 DATA "E",00,00,00,00,00,00,0
421 DATA "F",00,00,00,00,00,00,0
422 DATA "G",00,00,00,00,00,00,0
423 DATA "H",00,00,00,00,00,00,0
424 DATA "I",00,00,00,00,00,00,0

```

[illegible]

```

1000 1677: DRAIN -1.1: PLOT UAL "24
1001 1678: DRAIN UAL "107" DRAIN 1.1
1002 1679: GO SUB 6: IF 6=0:
1003 1680: THEN RETURN
1004 1681: IF 6=1: 2" AND 6=1: 2" THEN
1005 1682: LET 6=3: GO SUB 6: GO TO UAL
1006 1683: "1680"
1007 1684: GO SUB 6: DRAIN 1: RETURN
1008 1685: GO TO UAL "1680"
1009 1686: LET 6=2: GO SUB 1: GO SUB
1010 1687: 2
1011 1688: IF 6=1: THEN RETURN
1012 1689: IF 6=1: 2" THEN LET 6=3: GO
1013 1690: SUB 6: GO TO UAL "1680"
1014 1691: DRAIN 6: LET 6=3: GO SUB
1015 1692: 6
1016 1693: GO TO UAL "1680"
1017 1694: DRAIN 6: GO SUB 6: IF 6=0:
1018 1695: THEN RETURN
1019 1696: IF 6=1: 2" AND 6=1: 2" AND
1020 1697: 6=1: 2" THEN GO TO UAL "1680"
1021 1698: IF 6=1: 2" THEN LET 6=3: GO
1022 1699: TO UAL "1680"
1023 1700: RETURN
1024 1701: IF 6=2: OR 6=3: THEN GO
1025 1702: TO UAL "1680"
1026 1703: DRAIN 6: RETURN
1027 1704: DRAIN 6: DRAIN 6: DRAIN 6:
1028 1705: DRAIN 6: DRAIN 6: DRAIN 6:
1029 1706: DRAIN 6: DRAIN 6: DRAIN 6:
1030 1707: DRAIN 6: DRAIN 6: DRAIN 6:
1031 1708: DRAIN 6: DRAIN 6: DRAIN 6:
1032 1709: DRAIN 6: DRAIN 6: DRAIN 6:
1033 1710: DRAIN 6: DRAIN 6: DRAIN 6:
1034 1711: DRAIN 6: DRAIN 6: DRAIN 6:
1035 1712: DRAIN 6: DRAIN 6: DRAIN 6:
1036 1713: DRAIN 6: DRAIN 6: DRAIN 6:
1037 1714: DRAIN 6: DRAIN 6: DRAIN 6:
1038 1715: DRAIN 6: DRAIN 6: DRAIN 6:
1039 1716: DRAIN 6: DRAIN 6: DRAIN 6:
1040 1717: DRAIN 6: DRAIN 6: DRAIN 6:
1041 1718: DRAIN 6: DRAIN 6: DRAIN 6:
1042 1719: DRAIN 6: DRAIN 6: DRAIN 6:
1043 1720: DRAIN 6: DRAIN 6: DRAIN 6:
1044 1721: DRAIN 6: DRAIN 6: DRAIN 6:
1045 1722: DRAIN 6: DRAIN 6: DRAIN 6:
1046 1723: DRAIN 6: DRAIN 6: DRAIN 6:
1047 1724: DRAIN 6: DRAIN 6: DRAIN 6:
1048 1725: DRAIN 6: DRAIN 6: DRAIN 6:
1049 1726: DRAIN 6: DRAIN 6: DRAIN 6:
1050 1727: DRAIN 6: DRAIN 6: DRAIN 6:
1051 1728: DRAIN 6: DRAIN 6: DRAIN 6:
1052 1729: DRAIN 6: DRAIN 6: DRAIN 6:
1053 1730: DRAIN 6: DRAIN 6: DRAIN 6:
1054 1731: DRAIN 6: DRAIN 6: DRAIN 6:
1055 1732: DRAIN 6: DRAIN 6: DRAIN 6:
1056 1733: DRAIN 6: DRAIN 6: DRAIN 6:
1057 1734: DRAIN 6: DRAIN 6: DRAIN 6:
1058 1735: DRAIN 6: DRAIN 6: DRAIN 6:
1059 1736: DRAIN 6: DRAIN 6: DRAIN 6:
1060 1737: DRAIN 6: DRAIN 6: DRAIN 6:
1061 1738: DRAIN 6: DRAIN 6: DRAIN 6:
1062 1739: DRAIN 6: DRAIN 6: DRAIN 6:
1063 1740: DRAIN 6: DRAIN 6: DRAIN 6:
1064 1741: DRAIN 6: DRAIN 6: DRAIN 6:
1065 1742: DRAIN 6: DRAIN 6: DRAIN 6:
1066 1743: DRAIN 6: DRAIN 6: DRAIN 6:
1067 1744: DRAIN 6: DRAIN 6: DRAIN 6:
1068 1745: DRAIN 6: DRAIN 6: DRAIN 6:
1069 1746: DRAIN 6: DRAIN 6: DRAIN 6:
1070 1747: DRAIN 6: DRAIN 6: DRAIN 6:
1071 1748: DRAIN 6: DRAIN 6: DRAIN 6:
1072 1749: DRAIN 6: DRAIN 6: DRAIN 6:
1073 1750: DRAIN 6: DRAIN 6: DRAIN 6:
1074 1751: DRAIN 6: DRAIN 6: DRAIN 6:
1075 1752: DRAIN 6: DRAIN 6: DRAIN 6:
1076 1753: DRAIN 6: DRAIN 6: DRAIN 6:
1077 1754: DRAIN 6: DRAIN 6: DRAIN 6:
1078 1755: DRAIN 6: DRAIN 6: DRAIN 6:
1079 1756: DRAIN 6: DRAIN 6: DRAIN 6:
1080 1757: DRAIN 6: DRAIN 6: DRAIN 6:
1081 1758: DRAIN 6: DRAIN 6: DRAIN 6:
1082 1759: DRAIN 6: DRAIN 6: DRAIN 6:
1083 1760: DRAIN 6: DRAIN 6: DRAIN 6:
1084 1761: DRAIN 6: DRAIN 6: DRAIN 6:
1085 1762: DRAIN 6: DRAIN 6: DRAIN 6:
1086 1763: DRAIN 6: DRAIN 6: DRAIN 6:
1087 1764: DRAIN 6: DRAIN 6: DRAIN 6:
1088 1765: DRAIN 6: DRAIN 6: DRAIN 6:
1089 1766: DRAIN 6: DRAIN 6: DRAIN 6:
1090 1767: DRAIN 6: DRAIN 6: DRAIN 6:
1091 1768: DRAIN 6: DRAIN 6: DRAIN 6:
1092 1769: DRAIN 6: DRAIN 6: DRAIN 6:
1093 1770: DRAIN 6: DRAIN 6: DRAIN 6:
1094 1771: DRAIN 6: DRAIN 6: DRAIN 6:
1095 1772: DRAIN 6: DRAIN 6: DRAIN 6:
1096 1773: DRAIN 6: DRAIN 6: DRAIN 6:
1097 1774: DRAIN 6: DRAIN 6: DRAIN 6:
1098 1775: DRAIN 6: DRAIN 6: DRAIN 6:
1099 1776: DRAIN 6: DRAIN 6: DRAIN 6:
1100 1777: DRAIN 6: DRAIN 6: DRAIN 6:
1101 1778: DRAIN 6: DRAIN 6: DRAIN 6:
1102 1779: DRAIN 6: DRAIN 6: DRAIN 6:
1103 1780: DRAIN 6: DRAIN 6: DRAIN 6:
1104 1781: DRAIN 6: DRAIN 6: DRAIN 6:
1105 1782: DRAIN 6: DRAIN 6: DRAIN 6:
1106 1783: DRAIN 6: DRAIN 6: DRAIN 6:
1107 1784: DRAIN 6: DRAIN 6: DRAIN 6:
1108 1785: DRAIN 6: DRAIN 6: DRAIN 6:
1109 1786: DRAIN 6: DRAIN 6: DRAIN 6:
1110 1787: DRAIN 6: DRAIN 6: DRAIN 6:
1111 1788: DRAIN 6: DRAIN 6: DRAIN 6:
1112 1789: DRAIN 6: DRAIN 6: DRAIN 6:
1113 1790: DRAIN 6: DRAIN 6: DRAIN 6:
1114 1791: DRAIN 6: DRAIN 6: DRAIN 6:
1115 1792: DRAIN 6: DRAIN 6: DRAIN 6:
1116 1793: DRAIN 6: DRAIN 6: DRAIN 6:
1117 1794: DRAIN 6: DRAIN 6: DRAIN 6:
1118 1795: DRAIN 6: DRAIN 6: DRAIN 6:
1119 1796: DRAIN 6: DRAIN 6: DRAIN 6:
1120 1797: DRAIN 6: DRAIN 6: DRAIN 6:
1121 1798: DRAIN 6: DRAIN 6: DRAIN 6:
1122 1799: DRAIN 6: DRAIN 6: DRAIN 6:
1123 1800: DRAIN 6: DRAIN 6: DRAIN 6:
1124 1801: DRAIN 6: DRAIN 6: DRAIN 6:
1125 1802: DRAIN 6: DRAIN 6: DRAIN 6:
1126 1803: DRAIN 6: DRAIN 6: DRAIN 6:
1127 1804: DRAIN 6: DRAIN 6: DRAIN 6:
1128 1805: DRAIN 6: DRAIN 6: DRAIN 6:
1129 1806: DRAIN 6: DRAIN 6: DRAIN 6:
1130 1807: DRAIN 6: DRAIN 6: DRAIN 6:
1131 1808: DRAIN 6: DRAIN 6: DRAIN 6:
1132 1809: DRAIN 6: DRAIN 6: DRAIN 6:
1133 1810: DRAIN 6: DRAIN 6: DRAIN 6:
1134 1811: DRAIN 6: DRAIN 6: DRAIN 6:
1135 1812: DRAIN 6: DRAIN 6: DRAIN 6:
1136 1813: DRAIN 6: DRAIN 6: DRAIN 6:
1137 1814: DRAIN 6: DRAIN 6: DRAIN 6:
1138 1815: DRAIN 6: DRAIN 6: DRAIN 6:
1139 1816: DRAIN 6: DRAIN 6: DRAIN 6:
1140 1817: DRAIN 6: DRAIN 6: DRAIN 6:
1141 1818: DRAIN 6: DRAIN 6: DRAIN 6:
1142 1819: DRAIN 6: DRAIN 6: DRAIN 6:
1143 1820: DRAIN 6: DRAIN 6: DRAIN 6:
1144 1821: DRAIN 6: DRAIN 6: DRAIN 6:
1145 1822: DRAIN 6: DRAIN 6: DRAIN 6:
1146 1823: DRAIN 6: DRAIN 6: DRAIN 6:
1147 1824: DRAIN 6: DRAIN 6: DRAIN 6:
1148 1825: DRAIN 6: DRAIN 6: DRAIN 6:
1149 1826: DRAIN 6: DRAIN 6: DRAIN 6:
1150 1827: DRAIN 6: DRAIN 6: DRAIN 6:
1151 1828: DRAIN 6: DRAIN 6: DRAIN 6:
1152 1829: DRAIN 6: DRAIN 6: DRAIN 6:
1153 1830: DRAIN 6: DRAIN 6: DRAIN 6:
1154 1831: DRAIN 6: DRAIN 6: DRAIN 6:
1155 1832: DRAIN 6: DRAIN 6: DRAIN 6:
1156 1833: DRAIN 6: DRAIN 6: DRAIN 6:
1157 1834: DRAIN 6: DRAIN 6: DRAIN 6:
1158 1835: DRAIN 6: DRAIN 6: DRAIN 6:
1159 1836: DRAIN 6: DRAIN 6: DRAIN 6:
1160 1837: DRAIN 6: DRAIN 6: DRAIN 6:
1161 1838: DRAIN 6: DRAIN 
```

[illegible]

THE FABULOUS CASSETTE

50

FROM **Cascade**

VALUE that's out of this world

ONLY
£9.95
(including postage)

50 GAMES ON ONE CASSETTE

DRAGON DIB 1.1 Spectrum  Atari ORICA 2.0/4.0 VIO 2.0

It is impossible to tell you everything about the 50 games on CASSETTE 50 but they include many games such as: Snake, Wizard, Asterix, Tetris and many more. All the games are well tested, loaded in computer format, allowing you to play them on CASSETTE 50 with a single cassette. All the games will provide many hours of entertainment for all the family at a fraction of the cost of other computer games.

EXPRESS DELIVERY - ORDER NOW

Name
Address

Post Code

Country
Dealers & Stockists enquiries welcome.

Please send me by return of post, Cassette 50 at £9.95 per tape. I enclose a cheque/postal order for

£ made payable to Cascade Games Ltd.

Please debit my ☐ No.

SPECTRUM ☐ DRAGON ☐ ORICA ☐ VIO ☐

DRAGON ☐ ORICA ☐ VIO ☐

 Cascade Games Ltd, Suite 4, 1-31 Rensley Way, Hemmings, North, Yorkshire, HA1 5AB, England. Telephone: 04423 504115.

PG 1/85

campbell systems

Masterfile

YES - Microdrive compatible!

Spectrum 48K

As featured in the definitive filing system for the 48K Spectrum - MASTERFILE's machine coded flexibility gives you 200,000 (max) per file - 26 fields per record - up to 128 characters per field - multiple level searches for numeric or character stringing - data presentation in any one of 26 user-defined displays which may be imported by any field - USERBASIC for screen processing. the most comprehensive of the data-base

Revised June 1983

With example file and detailed manual £15.00

Drawmaster

Spectrum 48K

DRAWMASTER is the ultimate DRAW AND PAINT utility for the Spectrum - complete your own H&R pictures with 8 direction 3-speed fast draw keys - fast fill - comprehensive facility for all or part of the screen plus all the Spectrum colours and attributes - over 50 commands in all.

With detailed instructions £5.95

DLAN DISPLAY LANGUAGE

Spectrum 16/48K

DLAN (Draw and Display Language) is a powerful machine coded interpreter for generating BINARY TEXTUAL DISPLAYS for advertising and education - all commands are single letter or symbol - with 4-way sort of 11 type-faces through user-defined windows.

With detailed manual £7.95

All programs supplied 100 lines by return. Please include S&P and postage within Europe. D&P for full rate.

Campbell Systems Ltd, 10 Plover Road, Southport, Merseyside, L35 9BB, England. 07 524 0585

CS

machine code sort in
your Spectrum BASIC

ALL-SORT
simple to use
brilliant

£9.95

Write for literature & stockists:
Alan Firminger
171 Heme Hill, London SE24

CIRCUIT SKETCH



This incredible program will allow Spectrum users to make full use of their machine's high resolution graphics to form diagrams of electronic apparatus and either SAVE the results on tape or PRINT them out on the ZX Printer. An example of the resolution possible with this listing is shown in Fig. 1, an outdated radio circuit.

DRAWING ON EXPERIENCE

When RUN, the user will be asked to type in the numbers corresponding to the INK and PAPER colours required. An initial point will also be asked for; this is the point from which the sketch is to begin from.

The drawing of the circuit can then begin. To draw the components, the key corresponding to the first letter of the component should be pressed. The following represents a list of components included in the program:

Non-polarised capacitor — V
Variable capacitor — V

Electrolytic capacitor — V
Fixed resistor — Y
Variable resistor — Y
PNP transistor — Y
Diode — V
Inductor — Y
Switch — Y
Battery — W
Fuse — Y

Other components could easily be added should you require them.

Once you have pressed a certain key, the computer

Turn on and tune in for a bit of electronic 'circuit training'.

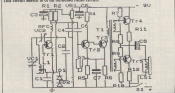
requires other information as to the nature of the component. For example, information must be given as to whether it is to be drawn vertically or horizontally; whether it should be drawn from the positive or negative end; whether a resistor is variable or fixed, etc. Once a component has been drawn, you can draw the interlinking lines using the cursor keys, '5', '8', '7' and '9' to draw a line left, down, up or right respectively. The computer will automatically allow you to start drawing from the other side of the component, or in the case of a transistor, from the collector.

GOING UP . . .

Components may be drawn horizontally or vertically (except the transistor), but the following rules must be noted:

- 1 — Horizontal components are drawn left to right.
- 2 — Vertical components are drawn from the bottom to the top.
- 3 — If you are drawing from left to right, the component must be horizontal.

Fig. 1. An example of the resolution available with this program. This circuit sketch is of an outdated radio circuit.



- 4 - Transistors are drawn horizontal, beginning at the base and ending up at the collector.
- 5 - Variable components are drawn from the central arrow position and terminate at the right or top end.

The current plotting position is always displayed; if you want to draw accurately or PRINT in component numbers or values of a later place, you would be

wise to note down the positions of each component.

You may not draw off the screen; you will be stopped by the contents of line 64. However, you are allowed to change the plotting position by pressing the 'p' key and stating the co-ordinates of the new position. The variable screen is set to zero if further drawing will result in the line going off the screen.

Once the circuit has been completed, you can press the

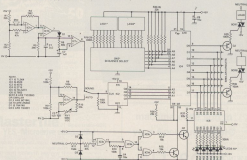
Break key and, using PRINT, mark in any additional script onto the diagram. Should you wish to SAVE the program, try using the following:

SAVE "chc0000" LINE 1

This will allow the program to RUN automatically when re-INSTALLED.

Should the thought of playing around with electronic circuits scare you to death, try using it as a simple stretched

A. Overall dimensions that were used in this research



Copyright © 2004 by John Wiley & Sons, Inc.

```

1  RE CIRCUIT DIAGRAM SKETCH
2  SPECIFIC: 2.2. INWARD 1000
3  INPUT: 2.2. INWARD 1000
4  INPUT: 2.2. INWARD 1000
5  INPUT: 2.2. INWARD 1000
6  INPUT: 2.2. INWARD 1000
7  INPUT: 2.2. INWARD 1000
8  INPUT: 2.2. INWARD 1000
9  INPUT: 2.2. INWARD 1000
10 INPUT: 2.2. INWARD 1000
11 INPUT: 2.2. INWARD 1000
12 INPUT: 2.2. INWARD 1000
13 INPUT: 2.2. INWARD 1000
14 INPUT: 2.2. INWARD 1000
15 INPUT: 2.2. INWARD 1000
16 INPUT: 2.2. INWARD 1000
17 INPUT: 2.2. INWARD 1000
18 INPUT: 2.2. INWARD 1000
19 INPUT: 2.2. INWARD 1000
20 INPUT: 2.2. INWARD 1000
21 INPUT: 2.2. INWARD 1000
22 INPUT: 2.2. INWARD 1000
23 INPUT: 2.2. INWARD 1000
24 INPUT: 2.2. INWARD 1000
25 INPUT: 2.2. INWARD 1000
26 INPUT: 2.2. INWARD 1000
27 INPUT: 2.2. INWARD 1000
28 INPUT: 2.2. INWARD 1000
29 INPUT: 2.2. INWARD 1000
30 INPUT: 2.2. INWARD 1000
31 INPUT: 2.2. INWARD 1000
32 INPUT: 2.2. INWARD 1000
33 INPUT: 2.2. INWARD 1000
34 INPUT: 2.2. INWARD 1000
35 INPUT: 2.2. INWARD 1000
36 INPUT: 2.2. INWARD 1000
37 INPUT: 2.2. INWARD 1000
38 INPUT: 2.2. INWARD 1000
39 INPUT: 2.2. INWARD 1000
40 INPUT: 2.2. INWARD 1000
41 INPUT: 2.2. INWARD 1000
42 INPUT: 2.2. INWARD 1000
43 INPUT: 2.2. INWARD 1000
44 INPUT: 2.2. INWARD 1000
45 INPUT: 2.2. INWARD 1000
46 INPUT: 2.2. INWARD 1000
47 INPUT: 2.2. INWARD 1000
48 INPUT: 2.2. INWARD 1000
49 INPUT: 2.2. INWARD 1000
50 INPUT: 2.2. INWARD 1000
51 INPUT: 2.2. INWARD 1000
52 INPUT: 2.2. INWARD 1000
53 INPUT: 2.2. INWARD 1000
54 INPUT: 2.2. INWARD 1000
55 INPUT: 2.2. INWARD 1000
56 INPUT: 2.2. INWARD 1000
57 INPUT: 2.2. INWARD 1000
58 INPUT: 2.2. INWARD 1000
59 INPUT: 2.2. INWARD 1000
60 INPUT: 2.2. INWARD 1000
61 INPUT: 2.2. INWARD 1000
62 INPUT: 2.2. INWARD 1000
63 INPUT: 2.2. INWARD 1000
64 INPUT: 2.2. INWARD 1000
65 INPUT: 2.2. INWARD 1000
66 INPUT: 2.2. INWARD 1000
67 INPUT: 2.2. INWARD 1000
68 INPUT: 2.2. INWARD 1000
69 INPUT: 2.2. INWARD 1000
70 INPUT: 2.2. INWARD 1000
71 INPUT: 2.2. INWARD 1000
72 INPUT: 2.2. INWARD 1000
73 INPUT: 2.2. INWARD 1000
74 INPUT: 2.2. INWARD 1000
75 INPUT: 2.2. INWARD 1000
76 INPUT: 2.2. INWARD 1000
77 INPUT: 2.2. INWARD 1000
78 INPUT: 2.2. INWARD 1000
79 INPUT: 2.2. INWARD 1000
80 INPUT: 2.2. INWARD 1000
81 INPUT: 2.2. INWARD 1000
82 INPUT: 2.2. INWARD 1000
83 INPUT: 2.2. INWARD 1000
84 INPUT: 2.2. INWARD 1000
85 INPUT: 2.2. INWARD 1000
86 INPUT: 2.2. INWARD 1000
87 INPUT: 2.2. INWARD 1000
88 INPUT: 2.2. INWARD 1000
89 INPUT: 2.2. INWARD 1000
90 INPUT: 2.2. INWARD 1000
91 INPUT: 2.2. INWARD 1000
92 INPUT: 2.2. INWARD 1000
93 INPUT: 2.2. INWARD 1000
94 INPUT: 2.2. INWARD 1000
95 INPUT: 2.2. INWARD 1000
96 INPUT: 2.2. INWARD 1000
97 INPUT: 2.2. INWARD 1000
98 INPUT: 2.2. INWARD 1000
99 INPUT: 2.2. INWARD 1000
100 INPUT: 2.2. INWARD 1000

```

[illegible]

[illegible]

BATS



In this program, you are a bat flying about trying to secure some moths for your supper. Movement is made via the four 'arrow' keys on the Spectrum, and to make a catch you have to place the centre of the bat character directly over the moth. To make it just a little bit more difficult, the moth moves totally at random.

Because your bat cannot fly in bright light, the moth can escape you by flying into the lighted window in the top left-hand corner of the screen. All you have to do is to trap the moth before it gets there! The game ends when you have attempted to catch ten moths, although there is a bonus if you manage to catch all ten.

HOLY BAT-CATCHER!

Initialisation of the graphics is done in lines 2230 to 2250. This leads straight into the instruction page which is completed by lines 2300 to 3100. This last section is not repeated when you play subsequent games after your first. The main program begins with line 50 which calls the subroutine to print the screen display, followed by lines 90 to 120 which initialise the screen co-ordinates for the bat and the moth. Lines 200 to 300 are a loop which constantly moves the position of the moth, and if the right keys are pressed this section of the program actually moves the bat.

Let this 16K program drive you batty!

The conditional statements for the moth make sure that it stays within the screen area and also reverse its direction should it get too near to the edge. The moth is moved 'dx' and 'dy' across and up the screen, but it is not incremented on each pass of the loop. It is the RND value in lines 230 and 240 which look after this latter task, and it is this which gives the moth its random movement.

The conditional statements for the bat read the keyboard, keep the bat on the screen and also ensure that it will not move into the area of the window. Line 320 detects a caught moth and moves on to the scoring section of the program. Lines 330 to 360 detect the moth in the vicinity of the window, and once there it is automatically drawn into the window, the moth's home score is updated and a new moth appears down in the bottom right-hand area of the screen.

Scoring is carried out from lines 370 to 500. This includes a 'high score' routine, and offers you a bonus 'go' if you managed to catch ten moths out of ten. This initiation comes courtesy of lines 530 to 580.

FLYING TONIGHT

The REM at line 20 indicates which lines need letters typed in the graphics mode. Lines 250 and 260 use keyboard graphics and it is important that you get in four blanks at the end of each string.

Line 3400 looks the most complicated. However, if you decide that you don't want to put a title or instructions in your listing, you could always re-write line 2400 to read:

2400 RETURN

and omit all from thereon.



```

5 REM *****
10 REM DATA ON C:\M\DOODCH
11 REM *****
20 REM 25 11025 255 388 378 25
25 THE LETTERS TO BE PRINTED ARE
THOSE LETTERS IN GRAPHICS MODE
30 RANDOMIZE CLS BORDER 1:
LET HSC=0 GO SUB 2000
35 HSC=0 CLS LET TET=0: L
ET ST=0: LET N=0
38 GO SUB 2000
40 LET N=10: LET N=20
42 LET Y=10+INT (RND*50) LET X
=0 IF RND>.4 THEN LET X=21
45 LET X=1: LET Y=1
48 REM MOVE BOTH
500 LET XX=X+1:YY=Y-1:XX=X+1:YY=Y
5 REM X,Y
510 LET XX=X+1:YY=Y-1:XX=X+1:YY=Y
520 PRINT AT Y,X: PAPER 5:GOTO 1
530 Y=Y+1
540 IF YY=0 THEN LET YY=40
550 IF XX=0 THEN LET XX=40
560 PRINT AT Y,X: PAPER 5:GOTO 1
7 AND Y=Y+1
570 REM MOVE BAT
575 PRINT AT Y,X:
580 LET Y=YY+1:INKEY$=" " AND
Y=YY-1:INKEY$=" " AND Y=YY+1
585 OR XX=X+1:INKEY$=" " AND
XX=X-1:INKEY$=" " AND XX=X+1
590 OR XX=X+1:INKEY$=" " AND XX=X-1
600 PRINT AT Y,X:GOTO 1
610 REM BOTH CAUGHT OR MOVE
620 IF XXX+1 AND Y=0 THEN GO
TO 370
630 IF Y=0 AND X=0 THEN LET XX=
-1: LET YY=-1
640 IF X=0 AND Y=1 THEN GO SUB
2000: LET Y=YY+1:PRINT AT 1,1
PAPER 7:GO IF X=0 THEN GO TO
610
650 IF X=0 AND Y=1 THEN GO TO 1
200
660 GO TO 200
670 REM BOTH CAUGHT
680 IF INKEY$="" THEN GO TO 38
3
690 LET ET=ET+1:PRINT AT 2,0:
DATA 7: PAPER 8: BRIGHT 1: NOTH
3:CAUGHT:LET PRU=100 IF ET:
=10 THEN GO TO 400
400 GO TO 100
410 REM +MOVE HOUND+
420 LET TET=TET+1 IF ET=10 ON
C HOUND THEN PRINT AT 10,0: PAPER
4: "IT'S A GOOD NIGHT FOR HUNTIN
42: "AT 17:10: PAPER 8: INK 7:
BRIGHT 1: FLASH 1: GOTO ON
IF INKEY$="" THEN GO TO 400
430 IF ET=10 HOUND=HOUND+1:PRU=
1000:FOR N=1 TO 10 STEP 2 GO
TO 440 TO 51:PRINT AT N,N: PAPER
5: "NEXT N: NEXT N: GO TO 10
4
440 IF TET=HSC THEN LET HSC=TET
450 PAPER 4: CLS PRINT AT 0,0
PAPER 6: "YOUR SCORE WAS
450 CLS "AT MOths EATS
4: "MO: MOths EATS
500 IF TET=10 THEN PRINT AT 7,0
PAPER 6: "YOUR TOTAL SCORE WAS
TET
510 IF ET=HSC THEN LET HSC=ET
520 PRINT AT 10,0: PAPER 3: INK
0: PAPER 7: GOTO 400
530 PRINT AT 1,1: PAPER 8: GO
TO 400
540 YOU MUST GO TO CONTINUE AT
14,4: "N=0 TO STOP OR "C=0

```

```

5030 "AT 15,4: " TO GO TO NEXT GO
5040 IF INKEY$="Y" OR INKEY$="N"
THEN CLS GO TO 40
5050 IF INKEY$="Y" OR INKEY$="N"
THEN GO TO 510
5060 IF INKEY$="C" OR INKEY$="N"
THEN CLS PRINT AT 10,10: PAPER
5: "PAPER 5: " START THE GAME
510 GO
520 GO TO 510
530 CLS FOR N=0 TO 31 STEP 2:
FOR M=0 TO 31 STEP 4: PRINT AT
N,M: "BD: " NEXT N: NEXT M
540 PRINT AT 10,1: INK 7: PAPER
1: BRIGHT 1: FLASH 1: "BYE BYE
5: "AT 14,10: PAPER 5: INVERSE
1: "END THE GAMES L: GO TO 0
550
5600 REM WINDOW
5610 LET N=0
5620 LET M=0
5630 FOR N=0 TO 7: FOR M=0 TO 7:
PRINT AT N,M: PAPER 5:
5640 NEXT N: NEXT M: FOR N=0 TO
3: PRINT AT N,M: PAPER 6: "NEA
T M: FOR M=0 TO 3 STEP 3: PRINT
AT N,M: PAPER 6: "NEA: NEXT N
5650 RETURN
5660 REM +HOUND+
5670 RESTORE 2010: DATA 0,0,100
5680 100,100,25,00
5690 REM +HOUND+
5700 DATA 0,0,0,0,0,100,0,100
5710 100,0,0,0,0,100,0,100
5720 0,0,0,0,0,100,0,100
5730 FOR N=0 TO 3: FOR M=0 TO 7:
PRINT AT N,M: PAPER 6: "NEA: NEXT N
5740 M=0: FOR N=0 TO 3: PRINT AT
N,M: "BD: " NEXT N
5750 FOR N=0 TO 31 STEP 2: FOR M
=0 TO 31 STEP 4: PRINT AT N,M:
5760 "BD: " NEXT N
5770 GO SUB 5400
5780 PRINT AT 0,0: "NE
5790 PRINT AT 14,3: PAPER 5: "PO
580 "ENTER" TO CONTINUE AT 2
581 11,1: BRIGHT 1: INK 0: FLASH 1:
582 11,1: BRIGHT 1: PAPER 8: CLS
583 PRINT AT 3,10: PAPER 3: INK
3: BRIGHT 1: FLASH 1: "EAT 3
5840 PRINT AT 4,0: "YOU ARE A BA
T HUNTING MOths "FOR SUPPER
5850 PRINT AT 0,0: "THE BAT IS M
OVED WITH KEYS A: " "S TO B-
5860 PRINT AT 0,0: "THE MOth IS
CAUGHT WHEN THE "BAT IS CENTRA
LLY OVER IT
5870 PRINT AT 10,0: "THE MOth IS
SAFE IN THE YELLOW "AREA OF
HE LIGHTED WINDOW" THE BAT CAN
NOT GO HERE"
5880 PRINT AT 10,0: "THE GAME EN
DS WITH 10 MOths: " "HOME" OR
CAUGHT
5890 PRINT AT 10,0: "PRESS "ENT
ER" TO CONTINUE " PAPER 6: CLS
5900 RETURN
5910 LET N=0
5920 RETURN

```

S B A T



LOTUS-SOFT



Remington-type joystick or keyboard.
100% machine code. Special
on-screen cursor graphics.
Strategic chess game.
Features. Avoid the
dodge, find the hidden exit and
escape. (3 star review).



GRAPHIC 194 by
Egg ball on perimeter, with
1 team, back. Features. Back.
Flashed Hawk, Laser, Miniball.
100% machine code. Program
size levels.



Graphic simulation of the
previous game of Hawks. Watch
the ball move towards the
goal. But have you chosen the
right and angle correctly?



Funniest educational game.
Full feature arithmetic. (4 star
review). Available in the full with
graphics. Also - Nine Lines
Calculator.

ALL CASSETTES AT ONLY £5.50 EACH, INCLUDING FIRST CLASS POST & PACKING

Send cheque/postal order to:
LOTUS-SOFT, 45 HADES ROAD, LLANGENHECH, LLANELLI, DYFED SAH 8UH

Name

Address

LORD HARRY
& LADY
HARRIET
BULL TABLES

Machine
Type
or
A

Total Enclosed

£

(BLOCK CAPITALS PLEASE)

See our games and much more at
POSE-TRON COMPUTING
3 STEPNEY ARCADE
LLANELLI (05542) 59624

Also at: REGAL SOFTWARE, DILLINGHAM
GAMES CENTRE, LONDON
and other good retail outlets

TRADE ENQUIRIES WELCOME

The No.1 Football Game

Football Manager

Designed by Kevin Toms

Some of the features of the game:

- Matches in 3D graphics
- Transfer market • Promotion and relegation • FA Cup matches • Injury problems • Full league tables • Four Divisions



- Pick your own team for each match
- As many seasons as you like
- Managerial rating
- 7 skill levels
- Save game facility.

• ZX81 Choice

Home Computing Weekly
14.00 (Jan 1985)

Addictive

Comments: Addictive game for ZX81 and our customers

who receive a free trial game on our ZX81.

Comments: Addictive game for ZX81 and our customers

who receive a free trial game on our ZX81.

Comments: Addictive game for ZX81 and our customers

who receive a free trial game on our ZX81.

Comments: Addictive game for ZX81 and our customers

who receive a free trial game on our ZX81.

Comments: Addictive game for ZX81 and our customers

who receive a free trial game on our ZX81.

ZX81 - FORTH ROM

with multi-tasking

Runs more than 10 tasks at once. Schedule tasks
to run from 50 times per second to once a year. Ideal
for control purposes. Three times faster than
FORTH, but 10 compatible. Available as a 16K-
yourself EPROM, with an extensive manual for £25
plus VAT. Some ready-converted ZX81's available.

Coming Soon!

FORTH-1/D cartridge for Spectrum £59 plus VAT.
Ask for details

David Husband

2 Gorseston Road, Branksome,
Poole BH12 1NW

Telephone: 0205 562145

From software products to hardware, we have it all.

From software products to hardware, we have it all.

From software products to hardware, we have it all.

From software products to hardware, we have it all.

From software products to hardware, we have it all.

From software products to hardware, we have it all.

From software products to hardware, we have it all.

3D NOUGHTS AND CROSSES

Add a new dimension to this old game in 48K.



This is a game played on a virtual board containing 27 spaces, any one of which can be filled with a nought or a cross. The board is represented on the screen by three conventional noughts and crosses boards placed one above the other. The levels are labelled A, B and C and the

spaces on each level are numbered from one to nine. The object of the game is not to make the first row of three — that is too easy for the player who goes first — but to make more complete rows than your opponent, the computer.

When the program is RUN, the computer asks you to choose

a skill level, then draws the board and asks if you want to go first. The computer plays the 'X' and you play the 'O'. Any row of three will count; some of them are hard to spot at first, but you'll soon get the hang of it. There are 49 possible rows. Examples: A1, B1 and C1 make a row, and so

do A1, B5 and C9. If you change your mind halfway through entering a move, press any obviously wrong key and you will be able to start your entry again.

In between each move the computer checks through all the possible rows and counts any completed ones by making an audible beep. The beep is higher for the computer's rows than for yours, so you can listen to your progress.

The computer is hard to beat at first, especially at level three, but it can be done, especially if you go first! Why not try to improve the computer's game? You can see how its strategy is organised by reading the program notes and you could, for example, change the order in which it checks for almost complete rows by changing the order of the numbers 1-49 in the DATA statements at lines 1060-1080.

VARIABLES

To save on memory use, most of the variables are only temporary, and are used for different purposes in different parts of the program. These are

- a\$ — Contains the character of the key last pressed when an input was requested.
- a1 — An array containing one element for each of the 27 spaces on the board.
- b0 — An array containing one element for each of the 49 possible wins at three.
- level — The skill level which has been selected.
- pos — The number of the element of a0 corresponding to the move which has been selected by you or the computer.
- go — The number of moves which have been made.
- m — An indicator of whose turn it is to move.

Other variables used are: g, l, a, b, p, q, t and so.

the variables which have a single constant use.

MEMORY

As listed, this program will not run on a 16K machine — it uses more than 8,300 bytes. However, it could be compressed for a 16K machine by omitting the REM statements and perhaps using some of the standard Sinclair memory saving methods. For example, numbers can be referred to using the VAL function, eg. PRINT AT VAL"22",VAL"2"; and so on. Again, some of the lines can be run together with the use of colons.

PROGRAM NOTES

Line 5 — Calls the instructions subroutine at line 9900.

Line 20-70 — Define the 'X' and 'O' graphics characters. This short routine, with the appropriate DATA statements, can be used to define any number of characters up to the maximum of 21.

Line 80-90 — Initialise the arrays a() and b(), and the move

counter, and set the screen attributes for the game.

Line 95 — POKE 23658,8 sets Caps Lock at capitals, which simplifies map-trapping. POKE 23659,1 allows you to PRINT AT line 23.

Line 100-175 — Set up the board.

Lines 180-198 — Set who is to go first. Line 190 calls an INKEY\$ subroutine at line 9900 which stores the key pressed in a\$. Note line 194, the effect of the comma after a PRINT AT statement is to blank out half a line per comma in the current PAPER colour. m is set to zero when it is the computer's move, and one when it is your move. If the computer is to go first, the program jumps to 1000.

Lines 200-330 — Ask for your move, and map-trap the results (again returned in a\$). An acceptable move is turned into a number from one to 27 and stored in the variable, pos, for use later when displaying the move on the screen. The corresponding element of array a() is checked to see if it is empty; if it is, a four is inserted. (Four is used for human's moves, and one for the computer's moves.) The program then jumps to line 8000.

Lines 1000-7910 — These are used for the computer's moves.

Line 1005 — Lets you know that the computer is working out its move. After each stage of

LEVEL		Your move?	LEVEL		You have
A		Press 1-9	A		3
B		OK (B, C)	B		I have 7
C		Then number 1-9	C		Loss.

Screen illustrations from the system. © Sanyo and Casio.

Screen illustrations from the program, 3D Noughts and Crosses.

working, if the computer has not found an acceptable move it will add an extra full stop after 'I'M THINKING...' so that you can see how hard it has had to think!

Lines 1010-1025 If the centre space is empty, the computer will always choose to go there. The centre space corresponds to element 14 in array *a()*. Once a move has been chosen, the computer always jumps to line 8000. If the centre space has already been filled, the computer will move on to line 1030.

Line 1030 At skill level one, the computer skips the rest of its strategy and jumps to a routine at line 7000 which generates a random move.

Lines 1060-1080 Contain DATA which tells the computer the order in which it is to scan array *b()* for the next steps in its strategy.

Line 1105 At skill level two, the computer skips the next two steps of its strategy.

Lines 1110-1125 Each element of array *b()* contains the sum of the numbers held in three elements of array *a()* corresponding to a row of three spaces on the board. For example, *a()* plus *a()* plus *a()* corresponds to the state of the row from the top far left of the board (space A1) to the top far right (space A3). This total is stored in one of the elements of array *b()*. In this section of the program, the computer checks through array *b()* looking for the number 18, which represents a row on the board containing two 'O's and an empty space. If the computer finds such a row, it jumps to a routine at line 7500 which will identify the empty space as that the computer can put an 'X' there to block your row.

Lines 1140-1175 If you are not in a position to complete a row next move, the computer checks through array *b()* again, this time looking for the number 7. This corresponds to a row on

the board containing two Xs and a blank space. If such a row is found, the computer again jumps to line 7500 so that it can identify the empty space and complete a row of Xs.

Lines 1180-1210 Failing these, the computer looks for a row containing two blank spaces and one X. If it finds one, it will need another X, threatening to complete a row on its next move.

Lines 1215-1265 If the computer can find none of these possibilities, it will check array *a()* in the order given in the DATA statement in line 1025 to see if any of the centre (also or corner) spaces are empty. It will make its move in the first one it finds. Plenty of space is left between lines 1265 and 7000 to insert more checks if you want to elaborate on the computer's strategy.

Lines 7000-7030 Generate a random move in any empty space on the board.

Lines 7500-7600 Contain a routine used by the computer to find which of the spaces, in a line it has identified is one of its scans of array *b()*, is empty. On exiting from this routine, the temporary variable, *i*, indicates the appropriate empty element in array *a()*.

Lines 7500 Put the computer's move into array *a()* and also into the variable, *pos*, for later use when displaying the move.

Lines 8000-8030 Find the correct screen address for the selected move and print an 'X' or 'O', as appropriate. The 'O' is a Graphics A and the 'X' is a Graphics B. The selected move is also displayed at the bottom of the screen.

Lines 8500-8745 Scan through array *a()* in all the combinations of three elements which correspond to a row of three on the board. In each case the sum of the three elements is stored in an element of array *b()* by means of a subroutine at line 9600. The temporary variable, *s*, is given the value 9500 to

shorten the repeated calls to this subroutine.

Lines 8750-8770 Increase the move counter by one and check for the end of the game.

Line 8810 Blanks out the message line at the bottom of the screen.

Lines 8820 If *s* is one, it becomes zero; if zero, it becomes one. This indicates whether it is your move or the computer's. This next line was borrowed from J.A. Ender's program, *Squareology*, in *2X Computing*, Summer 1982.

Line 9000 Jumps to line 1000 if it is the computer's move next, or line 200 if it is your move next.

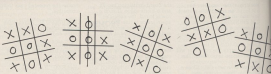
Lines 9000-9050 When the game is over, this routine counts the completed rows of Os and Xs by scanning array *b()*.

Lines 9200-9480 Display the final score and suitable messages, with jubilation or otherwise sound effects. (Room for improvement here, if you're interested.) You are asked if you want to play again. After the POKEs in line 95, if you break into direct command mode without POKE 23859,2 (line 9480) you will get a very strange screen display!

Lines 9500-9540 Contain an INKEY\$ subroutine which waits for a key to be pressed, and stores the result in a \$ for return.

Lines 9600-9640 Contain the subroutine used by the section of the program starting at line 8500. After each move, it stores the state of each possible row of three in an element of array *b()*. It gives a low blip when it finds a row of Os, and a higher one when it finds a row of Xs. It keeps quite about mixed or empty lines.

Lines 9900-9995 Contain the subroutine which displays the instructions and sets the skill level.



```

1 REM *****
2 REM
3 REM by Bill Welch, Mar 1980
4 REM
5 REM 3D NUGT
10 REM *****
20 FOR Q=144 TO 148: FOR I=0 TO 7
30 REM Q
40 REM USE CHRS 9-14
50 NEXT I: NEXT Q
60 DATA 0,128,132,138,68,30,38
70 DATA 0,136,72,56,24,20,12,1
80 REM *****
90 DIM A(127): DIM B(148): LET Q
100 BODDER 1: PAPER 1: INK 4: C
110 REM *****
120 REM *****
130 FOR I=1 TO 32 STEP 7
140 LET A=0: LET B=0
150 FOR P=1 TO 7:40 STEP 2
160 FOR Q=0 TO 5:40 STEP 4
170 PRINT AT 0,3: INK 0:CHRS A
180 LET A=A+1: NEXT Q
190 NEXT P: NEXT I
200 FOR I=1 TO 127 STEP 56
210 PLOT 100,I: DRAW -14,44
220 PLOT 100,I: DRAW -24,44
230 PLOT 100,I-14: DRAW 84,8
240 PLOT 100,I-14: DRAW 84,8
250 NEXT I
260 INK 0: POINT AT 0,0: "LEVEL
270 AT 0,2: "AT 10,0: "B" AT 17,2:
280 REM *****
290 REM *****
300 POINT AT 0,2: "DO YOU WANT
310 AT 0,3: "REPE .005,10
320 IF 1:10: REPE .005,10
330 IF 2:10: AND 3:10: THEN
340 "3-4: GO TO 100
350 IF AT 0,2: "
360 IF 3:10: THEN LET A=0: GO
370 TO 1000
380 LET A=1
390 REM *****
400 REM *****
410 POINT AT 0,2: "YOUR" AT 0,3:
420 "level" AT 12,2: "B" AT 17,2:
430 POINT AT 18,2: "level" AT 18,2:
440 "level" AT 22,2: "11-11"
450 REM *****
460 REM *****
470 IF 2:10: AND 3:10: AND
480 "0" THEN REPE .1:24: GO TO
490 1000
500 REPE .005,30
510 LET A=CODE 48-64
520 PRINT AT 22,0: "YOUR MOVE IS
530 AT 22,18:
540 GO TO 1000
550 IF 1:10: OR 3:10: THEN BE
560 .1:24: POINT "7 Start again,
570 GO TO 240
580 POINT AT 0,2: REPE .1:30
590 LET A=CODE 48-64
600 IF 2:10: AND 3:10: THEN REPE .1:
610 24: GO TO 1000
620 REM *****
630 REM *****
640 POINT AT 22,0: "I'M THINKING
650 REM *****
660 REM *****
670 IF 2:10: THEN LET A(14)=1
680 LET A=14: GO TO 1000
690 POINT AT 22,0: REPE .005,10
700 IF 2:10: THEN GO TO 1000
710 REM *****
720 DATA 11,21,5,12,28,18,17,30
730 30,39,36,48,43,48,37,30,31,34,4
740 DATA 28,1,12,9,10,19,7,3,10
750 31,37,18,39,44,39,20,43,30
760 DATA 11,30,18,17,2,8,4,26,2
770 3,8,13,21
780 REM *****
790 IF 1:10: THEN GO TO 1100
800 RESTORE 1000
810 FOR I=1 TO 40
820 REM I
830 IF 3:10: THEN GO TO 7500
840 NEXT I
850 PRINT " " REPE .005,10
860 REM *****
870 RESTORE 1000
880 FOR I=1 TO 40
890 REM I
900 IF 3:10: THEN GO TO 7500
910 NEXT I
920 PRINT " " REPE .005,10
930 REM *****
940 REM *****
950 POINT AT 0,2: "DO YOU WANT
960 AT 0,3: "REPE .005,10
970 REM *****
980 REM *****
990 POINT AT 0,2: "YOUR" AT 0,3:
1000 "level" AT 12,2: "B" AT 17,2:
1010 POINT AT 18,2: "level" AT 18,2:
1020 "level" AT 22,2: "11-11"
1030 REM *****
1040 REM *****
1050 IF 2:10: AND 3:10: AND
1060 "0" THEN REPE .1:24: GO TO
1070 1000
1080 REPE .005,30
1090 LET A=CODE 48-64
1100 PRINT AT 22,0: "YOUR MOVE IS
1110 AT 22,18:
1120 GO TO 1000
1130 IF 1:10: THEN FOR P=1 TO 14:
1140 STEP 2
1150 IF 1:10: AND 1:10: THEN LET A=
1160 11:20: AND 1:12: 1:11:20: AND 1:12:

```


[illegible]

AIR RAIDERS

Overhead excitement in this war-time simulation game with a difference.



This program differs a little bit from the normal 'war time' simulation games in that the shells actually follow a parabolic trajectory. The equation used to work out their motion is calculated in line 75 and simulates a projectile under gravity.

WHAT GOES UP...

The initial velocity and angle of

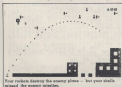
projection can either be selected by the player (mode 0) or by the computer (mode 1). The aim of the game is to destroy the overhead aircraft, launched ICBMs and the ICBM base; the ICBMs and their base must be destroyed using shells and the aircraft should be destroyed using rockets. The game stops either after eight cycles or if you accidentally destroy an allied aircraft (you can spot them because they're

black).

Full instructions are included in the program, including when you can fire a shell or a rocket. The program occupies virtually all of the 16K memory.

A VARIABLE SITUATION

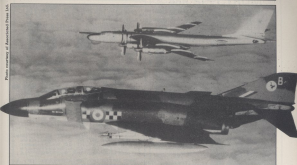
The variables used in this program are:



Four rockets destroy the enemy plane - but your shells missed the enemy missiles.

- q — The number of game cycles.
- p — The game mode.
- s — The score.
- r — The rocket drive.
- v — The rocket control.
- w — The random element in the aircraft and drive.
- z — The aircraft and ICBM drive.
- (x,y) — The shell trajectory pixels.
- (x,y) — The shell trajectory co-ordinates.
- z — The shell control.
- U/V on display — The shell launch velocity.
- A — The shell launch angle.

The variables used in the program.



```

1 100 DIM 1000
2 100 DIM 1000
3 100 DIM 1000
4 100 DIM 1000
5 100 DIM 1000
6 100 DIM 1000
7 100 DIM 1000
8 100 DIM 1000
9 100 DIM 1000
10 100 DIM 1000
11 100 DIM 1000
12 100 DIM 1000
13 100 DIM 1000
14 100 DIM 1000
15 100 DIM 1000
16 100 DIM 1000
17 100 DIM 1000
18 100 DIM 1000
19 100 DIM 1000
20 100 DIM 1000
21 100 DIM 1000
22 100 DIM 1000
23 100 DIM 1000
24 100 DIM 1000
25 100 DIM 1000
26 100 DIM 1000
27 100 DIM 1000
28 100 DIM 1000
29 100 DIM 1000
30 100 DIM 1000
31 100 DIM 1000
32 100 DIM 1000
33 100 DIM 1000
34 100 DIM 1000
35 100 DIM 1000
36 100 DIM 1000
37 100 DIM 1000
38 100 DIM 1000
39 100 DIM 1000
40 100 DIM 1000
41 100 DIM 1000
42 100 DIM 1000
43 100 DIM 1000
44 100 DIM 1000
45 100 DIM 1000
46 100 DIM 1000
47 100 DIM 1000
48 100 DIM 1000
49 100 DIM 1000
50 100 DIM 1000
51 100 DIM 1000
52 100 DIM 1000
53 100 DIM 1000
54 100 DIM 1000
55 100 DIM 1000
56 100 DIM 1000
57 100 DIM 1000
58 100 DIM 1000
59 100 DIM 1000
60 100 DIM 1000
61 100 DIM 1000
62 100 DIM 1000
63 100 DIM 1000
64 100 DIM 1000
65 100 DIM 1000
66 100 DIM 1000
67 100 DIM 1000
68 100 DIM 1000
69 100 DIM 1000
70 100 DIM 1000
71 100 DIM 1000
72 100 DIM 1000
73 100 DIM 1000
74 100 DIM 1000
75 100 DIM 1000
76 100 DIM 1000
77 100 DIM 1000
78 100 DIM 1000
79 100 DIM 1000
80 100 DIM 1000
81 100 DIM 1000
82 100 DIM 1000
83 100 DIM 1000
84 100 DIM 1000
85 100 DIM 1000
86 100 DIM 1000
87 100 DIM 1000
88 100 DIM 1000
89 100 DIM 1000
90 100 DIM 1000
91 100 DIM 1000
92 100 DIM 1000
93 100 DIM 1000
94 100 DIM 1000
95 100 DIM 1000
96 100 DIM 1000
97 100 DIM 1000
98 100 DIM 1000
99 100 DIM 1000
100 100 DIM 1000

```

Figure 1: A 2D grid world environment. The start cell is at (0,0) and the goal cell is at (3,3). There are obstacles at (1,1), (1,2), (2,1), and (2,2). The path from start to goal is (0,0) to (0,1) to (0,2) to (0,3) to (1,3) to (2,3) to (3,3).

ROSE SOFTWARE

EDUCATIONAL SOFTWARE FOR THE SPECTRUM (16K or 48K)

YOUNG LEARNERS II

4 programs for the young computer user, adding, subtracting, multiplying and dividing. All with excellent graphics. An easy introduction to computer aided learning.

£3.95 per cassette

G.C.E. "O" LEVEL MATHS REVISION PROGRAM

8 programs of multiple choice questions, covering properties of parallel lines, triangles, circles, polygons and circles. All questions are generated randomly. An explanation of the correct answer is given.

G.C.E. "O" LEVEL FRENCH VOCABULARY REVISION

Six programs on one cassette. The first 3 programs test vocabulary under various headings eg. in history. Third to fifth programs, further program, homework tests, common words and phrases, and irregular verbs.

We have a large range of educational programs for the Spectrum and DART.

Send me for catalogue of all our products to:
ROSE SOFTWARE

148 Welby Lane, Solihull, West Midlands B91 3LH

SPECTRUM

DUAL FUNCTION JOYSTICK INTERFACE

£13.95 inc

WHY PAY £25 FOR 2
INTERFACES WHEN YOU CAN
BUY OUR DUAL INTERFACE FOR
JUST £13.95 INC.

FEATURES

- Compatible with any software that lets the Kensington Joystick.
- Compatible with city software that uses cursor keys.
- Compatible with most Atari/Melbourne House games e.g. Kemp's Darts, Horace (slide timing etc).
- Allows this use of Atari/Melbourne House games e.g. Kemp's Darts, Horace (slide timing etc).
- Light indicators can be programmed in simple basic.
- Includes driver programmes and full instructions supplied.

STANDARD INTERFACE FEATURES

Same as Dual Interface

- But only compatible with any software that lets the Kensington Joystick.

ZX-81/SPECTRUM

LOADING PROGRAMS SOLVED WITH

the **COODER** as reviewed in the Spectator 1988

August '82

£7.95 INC.

DUAL INTERFACE £13.95 inc
STANDARD INTERFACE £10.95 inc

£1 discount if posted ordered with any interface
e.g. DUAL or S. Joystick £19.45 inc

Send Cheques/P.O. to:

ALEX ELECTRONICS

48 Parkside, Chalfont, Derby DE1 4GZ. Tel (0333) 700070

OVERSEAS £20.95 £22.95

48K SPECTRUM

1984

A GAME OF
GOVERNMENT MANAGEMENT



1984

THE GAME OF
ECONOMIC SURVIVAL

The 1984 Economy is in peril. AT THE CONTROL OF WHAT SORT OF CHANGELING MODEL YOU MEET WITH SEVERAL THOUSANDS TO SPEND & HOW PLANS TO THE NEXT GENERAL ELECTION? GRAPHIC DISPLAYS, NOT DEIGNED IN AN UGLY FOR ELEGANCE, BUT YES, AND ALL INCLUDED TO SHOW HOW YOU ARE DOING. HOW MANY YEARS WILL YOU LAST?

1984 (2000) Parker Bros. in
Reading House 1

£10.00

MOUNTAINS OF KET

A MONITOR OF AN ADVENTURE
PROGRAM CONSOLE (2000)

A MONITOR OF AN ADVENTURE
PROGRAM CONSOLE (2000)
A) THE MONS, MONS, MONS
EXTREME RACE, RACE, RACE
LEAD FACILITY PLUS MANY
OTHER FEATURES.

As well as being a fast response
compelling adventure it is the
Mountains of Ket is the best of a 3
part series that leads into a
mountain-adventure challenge.

Available in 2 parts for £10.00
if you order 100%.

£10.00

£10.00



SPLAT!



48K SPECTRUM CHALLENGE
FROM INCENTIVE SOFTWARE LTD

SPLAT!

AKADE

ONE OF THE MOST CHALLENGING
AND COMPLEX AND ANGLE
GAMES EVER PRODUCED
(TAKING OFF)

"SPLAT" is one of the most
challenging games I have ever played
on the 48K SPECTRUM. It is
certainly the most original"
Computer & Video Games

NON AVAILABLE FROM ANY OTHER
AND BOOKS.

£10.00

£10.00

All programs for the 48K SPECTRUM are available from all
good computer shops. In case of difficulty please order direct using
the coupon below.

Please send me (tick boxes) required)

1984 ☐ MOUNTAINS OF KET ☐ SPLAT ☐

All at £15.95 each (includes of VAT and tax data postage)

I enclose cheque/P.O. for £ or debit my Access Account by

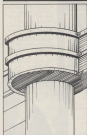


Name _____

Address _____

INCENTIVE SOFTWARE LTD., 34 London Street,
Reading RG1 4SL. Tel Reading 0734 997078

DRAINPIPES



Drainpipes is based on the old style of mechanical arcade game in which steel balls were dropped into cups or tubes.

The basis of the game is simple enough. A ball runs along a stepped routine at the top of the screen and will drop down if you press 'D'. Below it are a series of seven 'drainpipes' and you score by making it drop into the centre of one of these. Each drainpipe will only score once and a red indicator will light up to show where a score has been made. A score in each pipe will give a maximum of 28 points.

This version of the game has a random 'flip' feature. A red and white flag will sometimes illuminate as the ball is dropping and at the same time, the rate of drop slows down. If you don't like where the ball is heading press 'E' and it will slide off to the left and make its way back to the start without incrementing the ball score. You can then try again. Alternatively, as it slides off to the left you can press 'D' again whereupon it will once more go on its downward path. After 12 balls (a maximum score) the computer will tell you your

score and invite you to play again. If you don't want to continue, then pressing 'N' will get you a fond farewell.

DOWN THE DRAIN

The main movement of the ball and the setting of the graphics is done in lines 30 to 330. Lines 350 to 400 detect a scoring ball and label it with a score value. Lines 450 to 490 check to see if there is a full score line, in that no tube still has a score of zero, and if necessary terminates the round by allowing the next loop 500 to 530 to add up the score. These lines also add the score if you run out of balls.

Lines 600 to 660 print out the score and start a new game. Lines 670 to 690 terminate the game. The 'flip' routine is in lines 710 to 890 and shift the ball around the screen as required. Lines 900 onward print out the game instructions.

VARIATIONS ON A THEME

Anyone who wishes can find a lot of variations that can be done with this program. For a simple game, you can omit the 'flip' routine by leaving out line 750 and lines 710 to 890. You can also omit line 15 and the instructions.

Line 100 looks a little clumsy, but it works! It could be done with a subroutine of print statements. It can also be made

Here is an adaption of an old arcade game for you to try your hand at.

straight... though whether this makes for an easier or harder game is a matter of opinion.

You can make things a little more difficult by reducing 're' in line 150 thus getting less flips. You could also reduce the PAUSES in lines 730 and 800 to increase the speed of each ball.

And if you really want to get your fingers confused on the keyboard, why not write an extra subroutine called by INKEYS 'L' which will cause the ball to slide to the right. If you do this, the subroutine would be called from line 740.

A GRAPHIC EXPLANATION

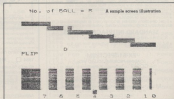
To help you type the program in, Colin has provided us with an explanation of some of the lines.

Line 100 includes seven Graphics shifted 8s followed by 31 spaces, then six Graphics shifted 8s followed by 31 spaces.

Line 340 includes four Graphics shifted 8s, one Graphics shifted 5, one Graphics 8, one Graphics 5, one Graphics shifted 5, and so on to until the end of the line which you should finish with a Graphics 5.

Line 360 includes one Graphics shifted 8.

Line 840 includes three Graphics shifted 8s.



MACHINE CODE COLOUR GRAPHICS

We show you how to speed up your colour graphics on the ZX Spectrum.



Although Spectrum BASIC is fairly fast, there are many occasions when it is not nearly fast enough, particularly when you need to print large or multiple images to the screen.

The most efficient way of solving this problem is to use machine code routines to do the job, for the speed of the Spectrum's microprocessor is such that several graphics, or even the entire screen, can be printed in a small fraction of a second. Although there is a routine in the ROM which enables you to do this (using an RST 10 instruction) the process can be rather long winded unless each character in the graphic follows the previous one on the screen. Furthermore, since this routine draws its graphics symbols from existing memory, you will always be

confined to using the standard character set and user defined graphics.

What would be ideal would be a program which enables you to 'read' every byte of your most complex graphic interferences and store them away in RAM for instant recall to any position on the screen. The two programs in this article, *Gratcode* and *Gratprint*, are designed to do exactly that.

DYNAMIC DUO

Gratprint is a machine code program which builds up a graphic image on the screen, including attributes, from a data file held in memory. It can either be called from within a BASIC program or incorporated in a larger machine code program where it can be used

to greater effect in generating laser fire or swooping aliens.

The *Demo* program illustrates *Gratprint* at work, printing a large green moon lander at the top left of the screen. By altering lines 20 and 30 which hold the low and high byte values of the first attribute position on the screen, you will see that the image can be printed anywhere, in spite of the peculiar design of the Sinclair display file.

Line 90 of *Demo* holds the machine code data of *Gratprint* and line 100 holds the data for the graphic.

The *Gratprint* program is designed to be loaded from address 32300 to address 32411 and therefore *RAMtop* should be at 32399 or less. The first eight bytes are used as temporary pigeon holes by the main program, which starts at

32308. Although Gnatprint can be moved elsewhere in memory, it contains references to these first eight bytes and these would therefore have to be changed.

Gnatcode is a BASIC program which generates data files of graphics for Gnatprint. To use it, first create your graphic image (or load) anywhere on the screen, preferably from the top left-hand corner where you can find the start address easily. The image may be created in the

be stored, the first attribute address currently occupied by the graphics and the number of character squares in the graphic. You will then be asked to input each of the displacements between the characters. These refer to attribute file displacements so that one square to the right = 1, one square below = 32, and so on. You may then sit back and turn a little tune for a few seconds whilst Gnatcode translates your handwork into a

Gnatprint works by moving a pointer around the attribute file of the Spectrum and loading the attributes of the current character into an address, followed by the corresponding eight display file bytes for each character. The data file which it reads is formatted so that each character is represented by 10 bytes. The first byte holds the displacement between the current character and the one before, the second holds the attribute code and the

Gnatprint Disassembled			
Addr	Hex Code	Mnemonic	
7E3C	00		7E38 2A327E LD HL,(7E38)
7E3D	98		7E39 110001 LD DE,1000h
7E3E	9C		7E3C A7 AND A,A
7E3F	7E		7E3D ED62 SBC HL,DE
7E40	000000		7E3F 1809 JR +09h
7E41	00		7E71 2A327E LD HL,(7E38)
7E42	2A2E7E	LD HL,(7E2Eh)	7E74 11000A LD DE,0A00h
7E43	03	EX DE,HL	7E77 A7 AND A,A
7E44	2A2E7E	LD HL,(7E2Eh)	7E78 ED62 SBC HL,DE
7E45	1A	LD A,(DE)	7E7A 01 POP DE
7E46	3E307E	LD (7E30h),A	7E7B 13 INC DE
7E47	13	INC DE	7E7C 0608 LD B,08h
7E48	1A	LD A,(DE)	7E7E 1A LD A,(DE)
7E49	1A	LD A,(DE)	7E7F 77 LD (HL),A
7E4A	77	LD (HL),A	7E80 06 DEC B
7E4B	D5	PUSH DE	7E81 2804 JR Z,+04h
7E4C	23327E	LD (7E3Ch),HL	7E83 24 INC H
7E4D	EB	EX DE,HL	7E84 13 INC DE
7E4E	21FF58	LD HL,58FFh	7E85 1807 JR -08h
7E4F	A7	AND A,A	7E87 2A327E LD HL,(7E38)
7E50	ED53	SBC HL,DE	7E8A 2A307E LD A,(7E30h)
7E51	3808	JR C,+08h	7E8D 060A LD B,0Ah
7E52	2A327E	LD HL,(7E38)	7E8F 90 SUB A,B
7E53	110018	LD DE,1800h	7E90 C8 RET Z
7E54	A7	AND A,A	7E92 3E307E LD (7E30h),A
7E55	ED62	SBC HL,DE	7E94 13 INC DE
7E56	1809	JR +09h	7E95 1A LD A,(DE)
7E57	ED6B327E	LD DE,(7E32h)	7E96 0600 LD B,00h
7E58	21FF59	LD HL,58FFh	7E98 4F LD C,A
7E59	A7	AND A,A	7E99 09 ADD HL,BC
7E5A	ED62	SBC HL,DE	7E9A 18A3 JR -3Ch
7E5B	3808	JR C,+08h	7E9C 0158FD LD BC,FD58h

Control Bytes

normal way, using BASIC colour commands, user defined graphics and so on. Then, when it is ready, activate Gnatcode by entering the command GO TO 9800. On no account RUN the program or your splendid new handwork will disappear without trace.

WHAT'S IN STORE?

You will then be prompted to input the address from which you want the graphics data to

sequential data file which can be accessed by Gnatprint.

Having loaded Gnatprint and prepared your data file, the next step is to load the start address of the data file into 32308/3 and the attribute address of the chosen screen location into 32309/1. Obviously in a moving graphics program, this would be done in machine code. To print the graphic to the screen, simply enter the command RANDOMIZE USR 32308.

remaining eight hold the graphic image. The data for the first character in a graphic also uses 10 bytes but instead of starting with a displacement value, it holds the total number of characters in the graphic.

The program incorporates a routine which calculates which of the three screen zones the current character will occupy and selects the corresponding display addresses accordingly. This ensures that if the graphic crosses a border between zones

then the relative positions of the characters are maintained.

THE SKY'S THE LIMIT

There is no limit to the size of the data file which can be used by Geoprint, other than the size of your machine's memory, and if files are built up section by section, there is no limit to the different shapes you can store. Having used 31 user-defined graphics in a picture you can convert it to data using Geocode, store it temporarily on tape and continue with a new set of graphics, bringing the whole lot together in one file when you have finished.

Graphics involving circles and other shapes can equally be stored.

If you intend to move graphics rapidly around the screen using Gnaprint, it will be necessary to erase each preceding image before printing the next. One way of doing this is to use a machine code equivalent of PRINT OVER, say by holding a blank graphic of the same size and shape as the original and using Gnaprint to print it over the top. In some cases, it will be sufficient to clear the screen between each printing because the speed of machine code is such that the illusion of continuous action will be

maintained. Screen clearing can be achieved by filling all the display file addresses with zero and all the attribute addresses with an appropriate code. If the background to your moving graphics is complex and you don't wish to clear it between moves, you can store a complete copy of the screen above RAMtop using a block move routine in machine code and reprint it to the screen between moves by means of a similar routine. Since this technique uses up nearly seven thousand bytes of RAM it is hardly worth contemplating on a HE Spectrum, unless the rest of your program is entirely in machine code.

Table 1

```

0000  REM  SAEFORME
0001  PRINT AT $1.0,"Enter data a
0002  INPUT data; LET salsdata=
0003  PRINT AT $1.0,"Enter first
0004  salsdata;
0005  all=0;
0006  POINT AT $1.0,"How many cas
0007  cases in graph?";
0008  INPUT char; DIM x(1); IF ch
0009  THEN DIM x(char-1);
0010  MORE data, char; LET salsdat
0011  =1;
0012  IF char=1 THEN GO TO 0008
0013  FOR x=1 TO char-1
0014  PRINT AT $1.0,"-
0015  GO
0016  PRINT AT $1.0,"Displacement
0017  ";
0018  INPUT x(1)
0019  NEXT x
0020  FOR x=1 TO char
0021  MORE data, MORE all
0022  IF all=0 THEN GO TO 0000
0023  IF all=0004 THEN GO TO 000
0024  LET salsall=sals
0025  GO TO 0000
0026  LET sals=all-0002
0027  GO TO 0000
0028  LET salsall=sals
0029  LET sals=0
0030  LET sals=sals+1
0031  MORE data, MORE sals
0032  LET sals=sals+1
0033  IF sals=1 THEN GO TO 0000
0034  LET sals=sals+000
0035  GO TO 0000
0036  LET sals=sals+1
0037  IF x(char) THEN MORE data, d
0038  IF x(char) THEN LET all=sals
0039  LET sals=sals+1
0040  MORE "
0041  PRINT AT $1.0,"Done-press ";
0042  GET
0043  STOP

```

[illegible]

© 2005 Blackwell Publishing Ltd, *Journal of Internal Medicine* 258: 101–108

The moon border graphic polished when the Census program is

[illegible]

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

ONLY THE BEST BECOME A... **FIGHTER PILOT**

A SPECTACULAR FLIGHT
SIMULATION OF THE WORLD'S
MOST EXCITING JET FIGHTER
WITH STUNNING 3D COCKPIT
VIEW AND THE MOST
REALISTIC 3D AIR-TO-AIR
COMBAT EVER SEEN ON
THE SPECTRUM!

The SUPREME SIMULATION.....

It leaves the rest at the
runway threshold!!

For more information on this and other games, visit our website at www.digitalintegration.co.uk or call 0800 123 4567. All prices include VAT and p&p (overseas add 50p per cassette). Trade & export enquiries welcome. All cassettes carry a lifetime guarantee.

£7.95

48k Spectrum
Atreslick compatible

DIGITAL Integration

Dept. 32 Ash Church Road, Ash, Aldershot, Hants GU11 6LR

All prices include VAT and p&p (overseas add 50p per cassette). Trade & export enquiries welcome.

Mail order or check your local computer store.

All cassettes carry a lifetime guarantee.

NIGHT GUNNER



NIGHT GUNNER - the air battle between you and the enemy. Fast action games for the Spectrum. Stunning graphics, exciting sound effects, and a challenge for all. All prices include VAT and p&p (overseas add 50p per cassette). Trade & export enquiries welcome. All cassettes carry a lifetime guarantee.

FOOTBALL



FOOTBALL - the ultimate challenge for the Spectrum. Fast action games for the Spectrum. Stunning graphics, exciting sound effects, and a challenge for all. All prices include VAT and p&p (overseas add 50p per cassette). Trade & export enquiries welcome. All cassettes carry a lifetime guarantee.

League Analysis

NEW

Taxi!



Taxi! - the ultimate challenge for the Spectrum. Fast action games for the Spectrum. Stunning graphics, exciting sound effects, and a challenge for all. All prices include VAT and p&p (overseas add 50p per cassette). Trade & export enquiries welcome. All cassettes carry a lifetime guarantee.

Take care of the variable though, if they wreck your stored you'll have to fork out \$20 to get the variable done!



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 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817

FIZZICAL

```

0340 IF A=0 THEN LET A=0
0350 CLS : PRINT AT 1.0: "SALES R
0360
0370 PLOT 14.100: DRAW 00.0
0380 PRINT
0390 PRINT --- Money in the 1st
0400
0410 PRINT --- Underscore sales
0420
0430 PRINT --- RENT=2"/100
0440 PRINT --- PROFIT FOR THE 04
0450
0460 LET A=0.100-0.100
0470 IF A=0 THEN GO TO 0500
0480 LET A=0.100+1
0490 IF A=0.100 THEN GO TO 0500
0500 IF A=0.100 THEN LET A=0.1
0510 PRINT AT 0.0: " Press 2 key
0520 CONTINUE
0530 IF INKEY="" THEN GO TO 0500
0540 GO TO 0
0550 STOP
0560 SET GRAPHIC SET UP
0570 CLS
0580 ORDER 0: PAPER 1
0590 FOR A=10 TO 31
0600 PRINT AT A.0: "
0610
0620 NEXT A
0630 PRINT AT 7.11: PAPER 2: In-
0640 LEADONADE
0650
0660 INK 0: PRINT AT 0.0: "
0670
0680 FOR A=0 TO 12
0690 PRINT AT A.0: "AT A.22: "
0700
0710 NEXT A
0720 FOR A=10 TO 17
0730 PRINT AT A.0: "
0740
0750 NEXT A
0760 PRINT AT 0.10: "
0770 PRINT AT 14.10: PAPER 2: In-
0780 LEADONADE
0790
0800 INK 0
0810 PLOT 00.00: DRAW 1.-0
0820 PLOT 00.00: DRAW -1.-0
0830 PLOT 00.00: DRAW 0.0
0840 FOR A=0 TO 20
0850 PLOT 00.0: DRAW 0.0
0860
0870 PHASE 0
0880 NEXT A
0890 PLOT 100.01: DRAW 1.-0
0900 PLOT 110.01: DRAW -1.-0
0910 PLOT 100.01: DRAW 0.0
0920 FOR A=0 TO 20
0930 PLOT 100.0: DRAW 0.0
0940
0950 PHASE 0
0960 NEXT A
0970 PHASE 00
0980 RETURN
0990 PHASE 000 CLS
1000 PRINT AT 1.0: "FINAL REPORT"
1010
1020 PLOT 14.100: DRAW 00.0
1030 PRINT --- You tested for
1040 " money. Until you can out
1050
1060 PRINT --- You didn't do very
1070 " well. You are not a good
1080
1090 PRINT AT 0.0: " Press 'ENTE
1100 " for another test"
1110 IF INKEY="" THEN GO TO 0500

```

```

1020 CLS : GO TO 0
1030 PHASE 000
1040 CLS
1050 PRINT AT 1.0: "FINAL REPORT"
1060 PLOT 14.100: DRAW 00.0
1070 PRINT --- You completed your
1080 " business and now
1090 " profit of
1100
1110 PRINT AT 10.0: "TOTAL MONEY"
1120
1130 PRINT AT 11.0: "TRIPLE FLOAT"
1140
1150 PLOT 00.70: DRAW 100.0
1160 PRINT AT 10.0: "PROFIT"
1170
1180 PLOT 14.100: DRAW 100.0
1190 PRINT AT 0.0: " Press 'ENTE
1200 " for another test"
1210 IF INKEY="" THEN GO TO 0500
1220
1230 CLS : GO TO 0
1240 STOP
1250 CLS
1260 PRINT AT 1.0: "POLICE REPORT
1270
1280 PLOT 14.100: DRAW 100.0
1290 PRINT --- The police have
1300 " informed you that vandals have
1310 " damaged your property. The cost
1320 " of repairs is $100.
1330
1340 PRINT --- If you cannot pay
1350 " $100 bill, press 0. If you
1360 " can't pay, go your business to
1370 " the
1380 IF INKEY="" THEN GO TO 0500
1390
1400 IF INKEY="" THEN GO TO 0500
1410
1420 LET A=0-0000
1430 GO TO 0
1440 SET INSTRUCTIONS
1450 PRINT AT 1.0: "LEADONADE STA
1460
1470 PLOT 70.100: DRAW 114.0
1480 PRINT --- You have been
1490 " charged of a small amount
1500 " and stand in
1510 " for a week. You start off
1520 " $10 in your pocket and you
1530 " would try to
1540 " as much money
1550 " as possible.
1560 PRINT --- You are given the
1570 " for a
1580 " report for
1590 " $100 to
1600 " how many cups of
1610 " and
1620 " at what price they
1630 " sold it"
1640 PRINT --- Press 'ENTER'
1650 " to
1660 IF INKEY="" THEN GO TO 0500
1670
1680 RETURN
1690 PHASE 000
1700 CLS
1710 PRINT AT 1.0: "BROKE"
1720 PLOT 14.100: DRAW 10.0
1730 PRINT --- You finally broke
1740 " up
1750
1760 PRINT --- The police have
1770 " told you that the vandals
1780 " are still on
1790 " $100. Press 'ENTE
1800 " for another test"
1810 IF INKEY="" THEN GO TO 0500
1820
1830 GO TO 0

```

POKEING THE SPECTRUM DISPLAY

The organisation of the Spectrum display takes a bit of getting used to, but it is not as bad as it seems.

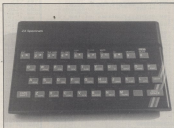
A lot of people must have felt a sudden cold chill after unpacking their brand new Spectrums and turning to page 184 of the user manual. Not only, it seems, has Mr. Sinclair decided to chop individual characters in the display file into tiny pieces and sprinkle them liberally over the screen but the screen itself has been divided into three separate blocks of eight lines each.

Before finding out whether things are really as bad as they seem I would like to expose you to the full horror of the situation. If you RUN Program 1, which POKES a byte of eight pixels into each consecutive address in the display file, you will see how the screen is built up.

Starting at the top left hand corner, the top row of each character square is filled in until the end of line seven is reached, when a jump is made to the second row of the first character in line zero, and so on. The process is then repeated with the second and third groups of eight lines until the screen is full.

SOLVING THE PROBLEM

As it turns out, the solution is not too difficult. In fact, once you have familiarised yourself with the display map, you will find that in some ways it is easier to handle than the one on the ZX81. In the first place, the display file starts at a fixed address and stays there rather than floating about in RAM. Secondly, there is no danger of POKING characters into hidden 'newline' flags which had the interesting effect on the earlier machine of destroying your



entire program when they were poked. Thirdly, since each character is composed of eight bytes, each of which is capable of holding up to eight pixels, or plot points, you can POKe any graphics you wish onto the screen without having to confine yourself to the 21 user definable graphics available in BASIC. Try running Program 1 again but this time substitute INTEND *255 for the number 255 and you will then get some idea of the infinite possibilities.

Suppose that you wished to POKe a 252 block of four black squares onto the screen near the top left hand corner. Program 2 illustrates clearly in BASIC how a machine code program might be designed to do this. The first address corresponds to the first row of the first character in line 1, column 1. The variable 'byte' is the number we wish to poke into this address and is read from the data statement in line 80. 255 corresponds to the binary number 11111111, which

will appear on the screen as a black bar. The 'displacement' variable, which is also read from the data statement, is the number of screen addresses which need to be added to the start address to position the first POKe, which in the first instance is zero. The second displacement is 1, which corresponds to the position immediately to the right of the start address, followed by 31 for the top row of the third character, 1 for the top of the fourth and then 223 to jump back to the second row of the first character and so on. The pause in line 50 is included to allow you to see how the whole graphic is built up.

DISPLACE HERE

It is necessary to do things in this order because every displacement value must be small enough to occupy a single byte in the data file. Although this does not matter in BASIC,

it is vital in our machine code programs because each item in the data table is handled one byte at a time.

Program 3 loads the machine code programs which will achieve exactly the effect as Program 2 but considerably faster. RAMTOP is set at 32229 and the program is loaded from 32300 onwards. The assembled program may then be executed by entering as a direct command: **RAND USR 32300**.

The data statement in line 70 holds the program itself whilst line 80 holds exactly the same data as in Program 2 except that the first number is 32 rather than zero, which is the total number of bytes in the complete graphic. Number 32 is initially loaded into address 32300 which during the execution of the program keeps a running total of the number of expected bytes remaining.

There is a valuable bonus gained by poking characters onto the screen by this method. Because each individual byte poked can correspond to any binary number between 00000000 and 11111111 and because each character is built up byte by byte, you can build up as many user definable graphics as you wish. Try experimenting by substituting other values for the 255s in Program 3.

BLOCK CAPITALS

Finally, as a little light relief, here is a simple method of poking 'block colour graphics' onto the screen.

The 'attributes' file in the Spectrum is happily arranged in a straightforward 32x24 format from address 22528 to 22595. By POKing any one of these addresses you can print squares of any colour and attribute anywhere on the screen. For example, any number between 32 and 39 which is POKed into an address will produce a green character square, whilst numbers between 96 and 103 will produce a bright green square.

Program 4 shows exactly which effects you will get as a result of POKing away

number between 0 and 255 into an attribute address. It should then be easy to adapt Program 3 to produce a machine code program which makes use of this to draw block graphics or test changing backgrounds.

```
LD HL, 16417  First address
LD DE, 32300  Data start
LD A, 128    Byte count
LD (32300), A
INC DE       Store
            Next data
```

```
LD A, (DE)   Byte code
LD (HL), A   Print
LD A, (32300)
LD B, 1
SUB B
RET Z        Finished?
LD (32300), A
INC DE       Next
            Displacement
LD A, (DE)
LD B, 0
LD C, A
ADD HL, BC   Next address
JP          Next data
```

THE PROGRAMS

```
1 REM Program 1
10 FOR a = 16384 TO 22607
20 POK a, 255
30 NEXT a
```

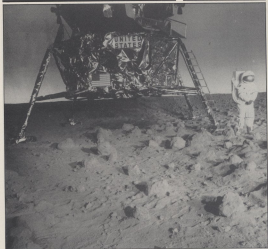
```
1 REM Program 2
2 BORDER 0
10 LET addr = 16417
20 READ displacement
30 READ byte
40 LET addr = addr + displacement
50 POK addr, byte
60 PAUSE 25
70 GO TO 20
80 DATA 0, 255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
```

```
1 REM Program 3
2 BORDER 0
10 CLEAR 32229
20 LET a = 32300
30 READ n
40 POK a, n
50 LET a = a + 1
60 GO TO 30
70 DATA 0, 31, 33, 64, 17, 77, 126, 26, 50, 44, 126, 19, 26,
    119, 28, 44, 126, 6, 1, 144, 200, 52, 44, 126, 19, 26,
    6, 0, 79, 9, 196, 55, 126
80 DATA 32, 255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
    255, 1, 255, 31, 255, 1, 255, 223
```

```
1 REM Program 4
10 LET byte = 0
20 FOR a = 22528 TO 22528 + 255
30 POK a, byte
40 LET byte = byte + 1
50 NEXT a
```

ASTEROIDS AHEAD

Steer clear of the asteroids and keep your spaceship flying.



There you are, a lonely spaceship floating through space, when suddenly your peace is shattered by a swarm of asteroids bearing down on you at a great rate of knots. You can maneuver your craft left using the 'J' key or right with the 'K' key. You also possess an energy field which you can use to deflect asteroids you cannot

avoid, but beware of running out of energy — the energy field has only a limited strength and a short range.

NO STONE UNTURNED

User-defined graphics have been used in this program for the asteroids, the energy screen, the spacecraft and the explosions. They are set during

the initial display, thus not causing an embarrassing pause during the game.

Logic lines have been used within the program. The most well-known example of these are present in line 50, but you may like to study the less popular conditional GOTO used in line 403.

Also, the use of ATTR to

AGF PROGRAMMABLE JOYSTICK INTERFACE

MICRODRIVE
COMPATIBLE

for
Spectrum
or **ZX81**



AGF PROGRAMMABLE INTERFACE

Developed as the only true hardware Programmable Joystick Interface, the product offers all the features associated with such a design.

You can use any Atari-compatible joystick controller with any software for your Atari Spectrum or ZX81, and just there with a joystick option.

Movement of the joystick is recognised by the computer exactly the same as pressing the appropriate control keys, and can therefore give the most immediate response to the development. The hardware programmed design works with all possible keymapping methods, both BASIC and Machine Code.

Eight directional movements, with or without the fire button being pressed, can be achieved by only programming the left, right, up, down and fire keys required by the game.

Programming is achieved by a two-digit code, which is looked up on the Programming Card supplied, for each direction and fire button. These two numbers are then selected on a pair of leads which are slipped onto appropriately numbered strips of the code tape.

Once configured this can be tested onto a Quick Reference Programming Card for checking with the game. As the programming is now power dependent the interface can be immediately used when and switched on.

The joystick remains fully functional and can be used simultaneously with the joystick.

An integral rear expansion connector means there is no need to fiddle the interface to connect other peripherals.

NB: A screen design improvement now means that the AGF Programmable Interface works with the new Outbladed II rapid "Auto Fire" feature.



KEY FEATURES

- Programmable design gives TOTAL software support.
- Sample: Alien, Competition Pro, Wux, Swampen, Yuck's Maze, Ltd track etc.
- Easy extension connector for all other add-ons.
- Free design program and instructions.

PACKAGE CONTENTS SUPPLIED

- Programmable Interface Module as described, complete with 40-pin programming leads.
- Self-adhesive programming chart detailing how to define which key is associated to UP, DOWN, LEFT, RIGHT, and FIRE.



- One pack of ten Quick Reference Programming Cards for at-machine setting to your game requirements.



- 12 months guarantee and full written instructions.

ONLY

16.95

+CT PER



Quickshot II Joystick

NEW IMPROVED GDB : BUILT IN STABILIZING SUCTION CUPS

TRIGGER FIRE BUTTON : RAMP AUTO FIRE SWITCH : TOP FIRE BUTTON

FROM: MR/MS/MRS

ADDRESS

SEND C.W.O. (NO STAMP NEEDED) TO: A.G.F. HARDWARE, DEPT.FS

FRANKFORD, BOOMER ROAD, WEST SUSSEX, PO22 8ER.

QTY	ITEM	ITEM PRICE	TOTAL
	PROGRAMMABLE INTERFACE	17.95	
	JOYSTICK(S)	17.95	
	PACK(S) QUICK REFERENCE CARDS	1.00	
☐ ZX81 ☐ ZX SPECTRUM ☐ Please tick		FINAL TOTAL	
SENDER ENGINEER WEL.COM		ADJUST PRICES ON APPLICATION	

LEARNER DRIVER

You'll have to watch out for more than the cat's eyes in this game.



A screen illustration.



A If you have to do in this program is to drive your car from the top left of the screen to the bottom right

without hitting anything. Sound's easy enough...

When you have got the program running, you find

yourself looking at a page of information explaining the rules of the game. Movement of your vehicle, which is placed at the top left-hand side of the screen, is controlled by the 'i' key to move up, the 'j' key to move down, the 'k' key to move left and the 'p' key to move right. Then, you have a minute period to maneuver your car to the bottom left-hand corner gaining as many points as you can.

MIND THAT BOLLARD

To gain points en route, you can demolish walls and fences, although you mustn't crash into a flashing bollard. Should you run over a pedestrian, or collide with one of the moving cars or trucks, you will also be told that you have crashed, given your score and asked if you would like to have another game.

You are permitted to run over cats, and as a special bonus you are awarded 100 points if you manage to run over the cat in the bottom right of the screen.

So, if you fancy an exciting drive, or you simply don't like cats, then this is the program for you.

```

1 INK 0: PAPER 0
2 LET HCOL=0
3 DATA 100,250,170,250,24,250
4 DATA 100,250,170,250,24,250
5 DATA 100,250,170,250,24,250
6 DATA 100,250,170,250,24,250
7 DATA 100,250,170,250,24,250
8 DATA 100,250,170,250,24,250
9 DATA 100,250,170,250,24,250
10 DATA 100,250,170,250,24,250
11 DATA 100,250,170,250,24,250
12 DATA 100,250,170,250,24,250
13 DATA 100,250,170,250,24,250
14 DATA 100,250,170,250,24,250
15 DATA 100,250,170,250,24,250
16 DATA 100,250,170,250,24,250
17 DATA 100,250,170,250,24,250
18 DATA 100,250,170,250,24,250
19 DATA 100,250,170,250,24,250
20 DATA 100,250,170,250,24,250
21 DATA 100,250,170,250,24,250
22 DATA 100,250,170,250,24,250
23 DATA 100,250,170,250,24,250
24 DATA 100,250,170,250,24,250
25 DATA 100,250,170,250,24,250
26 DATA 100,250,170,250,24,250
27 DATA 100,250,170,250,24,250
28 DATA 100,250,170,250,24,250
29 DATA 100,250,170,250,24,250
30 DATA 100,250,170,250,24,250
31 DATA 100,250,170,250,24,250
32 DATA 100,250,170,250,24,250
33 DATA 100,250,170,250,24,250
34 DATA 100,250,170,250,24,250
35 DATA 100,250,170,250,24,250
36 DATA 100,250,170,250,24,250
37 DATA 100,250,170,250,24,250
38 DATA 100,250,170,250,24,250
39 DATA 100,250,170,250,24,250
40 DATA 100,250,170,250,24,250
41 DATA 100,250,170,250,24,250
42 DATA 100,250,170,250,24,250
43 DATA 100,250,170,250,24,250
44 DATA 100,250,170,250,24,250
45 DATA 100,250,170,250,24,250
46 DATA 100,250,170,250,24,250
47 DATA 100,250,170,250,24,250
48 DATA 100,250,170,250,24,250
49 DATA 100,250,170,250,24,250
50 DATA 100,250,170,250,24,250
51 DATA 100,250,170,250,24,250
52 DATA 100,250,170,250,24,250
53 DATA 100,250,170,250,24,250
54 DATA 100,250,170,250,24,250
55 DATA 100,250,170,250,24,250
56 DATA 100,250,170,250,24,250
57 DATA 100,250,170,250,24,250
58 DATA 100,250,170,250,24,250
59 DATA 100,250,170,250,24,250
60 DATA 100,250,170,250,24,250
61 DATA 100,250,170,250,24,250
62 DATA 100,250,170,250,24,250
63 DATA 100,250,170,250,24,250
64 DATA 100,250,170,250,24,250
65 DATA 100,250,170,250,24,250
66 DATA 100,250,170,250,24,250
67 DATA 100,250,170,250,24,250
68 DATA 100,250,170,250,24,250
69 DATA 100,250,170,250,24,250
70 DATA 100,250,170,250,24,250
71 DATA 100,250,170,250,24,250
72 DATA 100,250,170,250,24,250
73 DATA 100,250,170,250,24,250
74 DATA 100,250,170,250,24,250
75 DATA 100,250,170,250,24,250
76 DATA 100,250,170,250,24,250
77 DATA 100,250,170,250,24,250
78 DATA 100,250,170,250,24,250
79 DATA 100,250,170,250,24,250
80 DATA 100,250,170,250,24,250
81 DATA 100,250,170,250,24,250
82 DATA 100,250,170,250,24,250
83 DATA 100,250,170,250,24,250
84 DATA 100,250,170,250,24,250
85 DATA 100,250,170,250,24,250
86 DATA 100,250,170,250,24,250
87 DATA 100,250,170,250,24,250
88 DATA 100,250,170,250,24,250
89 DATA 100,250,170,250,24,250
90 DATA 100,250,170,250,24,250
91 DATA 100,250,170,250,24,250
92 DATA 100,250,170,250,24,250
93 DATA 100,250,170,250,24,250
94 DATA 100,250,170,250,24,250
95 DATA 100,250,170,250,24,250
96 DATA 100,250,170,250,24,250
97 DATA 100,250,170,250,24,250
98 DATA 100,250,170,250,24,250
99 DATA 100,250,170,250,24,250
100 DATA 100,250,170,250,24,250

```

```

60 NEXT J
61 FOR J=0 TO 7
62 READ A
63 FOR K=0+120+J
64 NEXT K
65 FOR L=0 TO 7
66 READ B
67 FOR M=0+120+J
68 NEXT M
69 FOR N=0 TO 7
70 READ C
71 FOR O=0+120+J
72 NEXT O
73 FOR P=0 TO 7
74 READ D
75 FOR Q=0+120+J
76 NEXT Q
77 FOR R=0 TO 7
78 READ E
79 FOR S=0+120+J
80 NEXT S
81 FOR T=0 TO 7
82 READ F
83 FOR U=0+120+J
84 NEXT U
85 FOR V=0 TO 7
86 READ G
87 FOR W=0+120+J
88 NEXT W
89 FOR X=0 TO 7
90 READ H
91 FOR Y=0+120+J
92 NEXT Y
93 FOR Z=0 TO 7
94 READ I
95 FOR AA=0+120+J
96 NEXT AA
97 FOR AB=0 TO 7
98 READ J
99 FOR AC=0+120+J
100 NEXT AC

```

```

100 PRINT AT 3,0,"LEARNER DRIVER"
101 AT 3,0:GOTO 100
102 GOTO 100
103 GOTO 100
104 GOTO 100
105 GOTO 100
106 GOTO 100
107 GOTO 100
108 GOTO 100
109 GOTO 100
110 GOTO 100
111 GOTO 100
112 GOTO 100
113 GOTO 100
114 GOTO 100
115 GOTO 100
116 GOTO 100
117 GOTO 100
118 GOTO 100
119 GOTO 100
120 GOTO 100

```

```

11400.    AND P1=0 THEN BELLS=0
11410.    ALTHOUGH CARRIAGE OVER 240 IS 0
11420.    PRINT: THE FINAL 241 STOPS &
11430.    BELLS=0: GOTO 11440
11440.    OUT 255,INT (RND*(255)-PRUSE
11450.
11460.    PRINT AT 12.5,"TIME LEFT:
11470.    0 MIN."
11480.    PRINT AT 14.5,"CONTROLS UP
11490.    1 LEFT 0"
11500.    PRINT AT 16.15,"DOWN 1" 0
11510.    PRINT AT 18.7,"PRESS 1" 10
11520.    IF INKEY$="P" THEN GO TO 15
11530.    GO TO 120
11540.    CLS
11550.    DEF FN 11:INT (255*(RND-PRUSE
11560.    25574+255*PRUSE/25573+PRUSE/25573
11570.    )/255)
11580.    LET 11=FN 11
11590.    LET 11=0
11600.    LET 11=0
11610.    LET 11=0
11620.    PRINT DM 3,AT 17.5,"F" 10
11630.    "F" 10,AT 20.25,"F" 10,AT 22.5,
11640.    "F" 10,AT 25.25,"F" 10,AT 28.5,
11650.    "F" 10,AT 31.25,"F" 10,AT 34.5,
11660.    "F" 10,AT 37.5,"F" 10,AT 40.5,
11670.    "F" 10,AT 43.5,"F" 10,AT 46.5,
11680.    "F" 10,AT 49.5,"F" 10,AT 52.5,
11690.    "F" 10,AT 55.5,"F" 10,AT 58.5,
11700.    "F" 10,AT 61.5,"F" 10,AT 64.5,
11710.    "F" 10,AT 67.5,"F" 10,AT 70.5,
11720.    "F" 10,AT 73.5,"F" 10,AT 76.5,
11730.    "F" 10,AT 79.5,"F" 10,AT 82.5,
11740.    "F" 10,AT 85.5,"F" 10,AT 88.5,
11750.    "F" 10,AT 91.5,"F" 10,AT 94.5,
11760.    "F" 10,AT 97.5,"F" 10,AT 100.5,
11770.    "F" 10,AT 103.5,"F" 10,AT 106.5,
11780.    "F" 10,AT 109.5,"F" 10,AT 112.5,
11790.    "F" 10,AT 115.5,"F" 10,AT 118.5,
11800.    "F" 10,AT 121.5,"F" 10,AT 124.5,
11810.    "F" 10,AT 127.5,"F" 10,AT 130.5,
11820.    "F" 10,AT 133.5,"F" 10,AT 136.5,
11830.    "F" 10,AT 139.5,"F" 10,AT 142.5,
11840.    "F" 10,AT 145.5,"F" 10,AT 148.5,
11850.    "F" 10,AT 151.5,"F" 10,AT 154.5,
11860.    "F" 10,AT 157.5,"F" 10,AT 160.5,
11870.    "F" 10,AT 163.5,"F" 10,AT 166.5,
11880.    "F" 10,AT 169.5,"F" 10,AT 172.5,
11890.    "F" 10,AT 175.5,"F" 10,AT 178.5,
11900.    "F" 10,AT 181.5,"F" 10,AT 184.5,
11910.    "F" 10,AT 187.5,"F" 10,AT 190.5,
11920.    "F" 10,AT 193.5,"F" 10,AT 196.5,
11930.    "F" 10,AT 199.5,"F" 10,AT 202.5,
11940.    "F" 10,AT 205.5,"F" 10,AT 208.5,
11950.    "F" 10,AT 211.5,"F" 10,AT 214.5,
11960.    "F" 10,AT 217.5,"F" 10,AT 220.5,
11970.    "F" 10,AT 223.5,"F" 10,AT 226.5,
11980.    "F" 10,AT 229.5,"F" 10,AT 232.5,
11990.    "F" 10,AT 235.5,"F" 10,AT 238.5,
12000.    "F" 10,AT 241.5,"F" 10,AT 244.5,
12010.    "F" 10,AT 247.5,"F" 10,AT 250.5,
12020.    "F" 10,AT 253.5,"F" 10,AT 256.5,
12030.    "F" 10,AT 259.5,"F" 10,AT 262.5,
12040.    "F" 10,AT 265.5,"F" 10,AT 268.5,
12050.    "F" 10,AT 271.5,"F" 10,AT 274.5,
12060.    "F" 10,AT 277.5,"F" 10,AT 280.5,
12070.    "F" 10,AT 283.5,"F" 10,AT 286.5,
12080.    "F" 10,AT 289.5,"F" 10,AT 292.5,
12090.    "F" 10,AT 295.5,"F" 10,AT 298.5,
12100.    "F" 10,AT 301.5,"F" 10,AT 304.5,
12110.    "F" 10,AT 307.5,"F" 10,AT 310.5,
12120.    "F" 10,AT 313.5,"F" 10,AT 316.5,
12130.    "F" 10,AT 319.5,"F" 10,AT 322.5,
12140.    "F" 10,AT 325.5,"F" 10,AT 328.5,
12150.    "F" 10,AT 331.5,"F" 10,AT 334.5,
12160.    "F" 10,AT 337.5,"F" 10,AT 340.5,
12170.    "F" 10,AT 343.5,"F" 10,AT 346.5,
12180.    "F" 10,AT 349.5,"F" 10,AT 352.5,
12190.    "F" 10,AT 355.5,"F" 10,AT 358.5,
12200.    "F" 10,AT 361.5,"F" 10,AT 364.5,
12210.    "F" 10,AT 367.5,"F" 10,AT 370.5,
12220.    "F" 10,AT 373.5,"F" 10,AT 376.5,
12230.    "F" 10,AT 379.5,"F" 10,AT 382.5,
12240.    "F" 10,AT 385.5,"F" 10,AT 388.5,
12250.    "F" 10,AT 391.5,"F" 10,AT 394.5,
12260.    "F" 10,AT 397.5,"F" 10,AT 400.5,
12270.    "F" 10,AT 403.5,"F" 10,AT 406.5,
12280.    "F" 10,AT 409.5,"F" 10,AT 412.5,
12290.    "F" 10,AT 415.5,"F" 10,AT 418.5,
12300.    "F" 10,AT 421.5,"F" 10,AT 424.5,
12310.    "F" 10,AT 427.5,"F" 10,AT 430.5,
12320.    "F" 10,AT 433.5,"F" 10,AT 436.5,
12330.    "F" 10,AT 439.5,"F" 10,AT 442.5,
12340.    "F" 10,AT 445.5,"F" 10,AT 448.5,
12350.    "F" 10,AT 451.5,"F" 10,AT 454.5,
12360.    "F" 10,AT 457.5,"F" 10,AT 460.5,
12370.    "F" 10,AT 463.5,"F" 10,AT 466.5,
12380.    "F" 10,AT 469.5,"F" 10,AT 472.5,
12390.    "F" 10,AT 475.5,"F" 10,AT 478.5,
12400.    "F" 10,AT 481.5,"F" 10,AT 484.5,
12410.    "F" 10,AT 487.5,"F" 10,AT 490.5,
12420.    "F" 10,AT 493.5,"F" 10,AT 496.5,
12430.    "F" 10,AT 499.5,"F" 10,AT 502.5,
12440.    "F" 10,AT 505.5,"F" 10,AT 508.5,
12450.    "F" 10,AT 511.5,"F" 10,AT 514.5,
12460.    "F" 10,AT 517.5,"F" 10,AT 520.5,
12470.    "F" 10,AT 523.5,"F" 10,AT 526.5,
12480.    "F" 10,AT 529.5,"F" 10,AT 532.5,
12490.    "F" 10,AT 535.5,"F" 10,AT 538.5,
12500.    "F" 10,AT 541.5,"F" 10,AT 544.5,
12510.    "F" 10,AT 547.5,"F" 10,AT 550.5,
12520.    "F" 10,AT 553.5,"F" 10,AT 556.5,
12530.    "F" 10,AT 559.5,"F" 10,AT 562.5,
12540.    "F" 10,AT 565.5,"F" 10,AT 568.5,
12550.    "F" 10,AT 571.5,"F" 10,AT 574.5,
12560.    "F" 10,AT 577.5,"F" 10,AT 580.5,
12570.    "F" 10,AT 583.5,"F" 10,AT 586.5,
12580.    "F" 10,AT 589.5,"F" 10,AT 592.5,
12590.    "F" 10,AT 595.5,"F" 10,AT 598.5,
12600.    "F" 10,AT 601.5,"F" 10,AT 604.5,
12610.    "F" 10,AT 607.5,"F" 10,AT 610.5,
12620.    "F" 10,AT 613.5,"F" 10,AT 616.5,
12630.    "F" 10,AT 619.5,"F" 10,AT 622.5,
12640.    "F" 10,AT 625.5,"F" 10,AT 628.5,
12650.    "F" 10,AT 631.5,"F" 10,AT 634.5,
12660.    "F" 10,AT 637.5,"F" 10,AT 640.5,
12670.    "F" 10,AT 643.5,"F" 10,AT 646.5,
12680.    "F" 10,AT 649.5,"F" 10,AT 652.5,
12690.    "F" 10,AT 655.5,"F" 10,AT 658.5,
12700.    "F" 10,AT 661.5,"F" 10,AT 664.5,
12710.    "F" 10,AT 667.5,"F" 10,AT 670.5,
12720.    "F" 10,AT 673.5,"F" 10,AT 676.5,
12730.    "F" 10,AT 679.5,"F" 10,AT 682.5,
12740.    "F" 10,AT 685.5,"F" 10,AT 688.5,
12750.    "F" 10,AT 691.5,"F" 10,AT 694.5,
12760.    "F" 10,AT 697.5,"F" 10,AT 700.5,
12770.    "F" 10,AT 703.5,"F" 10,AT 706.5,
12780.    "F" 10,AT 709.5,"F" 10,AT 712.5,
12790.    "F" 10,AT 715.5,"F" 10,AT 718.5,
12800.    "F" 10,AT 721.5,"F" 10,AT 724.5,
12810.    "F" 10,AT 727.5,"F" 10,AT 730.5,
12820.    "F" 10,AT 733.5,"F" 10,AT 736.5,
12830.    "F" 10,AT 739.5,"F" 10,AT 742.5,
12840.    "F" 10,AT 745.5,"F" 10,AT 748.5,
12850.    "F" 10,AT 751.5,"F" 10,AT 754.5,
12860.    "F" 10,AT 757.5,"F" 10,AT 760.5,
12870.    "F" 10,AT 763.5,"F" 10,AT 766.5,
12880.    "F" 10,AT 769.5,"F" 10,AT 772.5,
12890.    "F" 10,AT 775.5,"F" 10,AT 778.5,
12900.    "F" 10,AT 781.5,"F" 10,AT 784.5,
12910.    "F" 10,AT 787.5,"F" 10,AT 790.5,
12920.    "F" 10,AT 793.5,"F" 10,AT 796.5,
12930.    "F" 10,AT 799.5,"F" 10,AT 802.5,
12940.    "F" 10,AT 805.5,"F" 10,AT 808.5,
12950.    "F" 10,AT 811.5,"F" 10,AT 814.5,
12960.    "F" 10,AT 817.5,"F" 10,AT 820.5,
12970.    "F" 10,AT 823.5,"F" 10,AT 826.5,
12980.    "F" 10,AT 829.5,"F" 10,AT 832.5,
12990.    "F" 10,AT 835.5,"F" 10,AT 838.5,
13000.    "F" 10,AT 841.5,"F" 10,AT 844.5,
13010.    "F" 10,AT 847.5,"F" 10,AT 850.5,
13020.    "F" 10,AT 853.5,"F" 10,AT 856.5,
13030.    "F" 10,AT 859.5,"F" 10,AT 862.5,
13040.    "F" 10,AT 865.5,"F" 10,AT 868.5,
13050.    "F" 10,AT 871.5,"F" 10,AT 874.5,
13060.    "F" 10,AT 877.5,"F" 10,AT 880.5,
13070.    "F" 10,AT 883.5,"F" 10,AT 886.5,
13080.    "F" 10,AT 889.5,"F" 10,AT 892.5,
13090.    "F" 10,AT 895.5,"F" 10,AT 898.5,
13100.    "F" 10,AT 901.5,"F" 10,AT 904.5,
13110.    "F" 10,AT 907.5,"F" 10,AT 910.5,
13120.    "F" 10,AT 913.5,"F" 10,AT 916.5,
13130.    "F" 10,AT 919.5,"F" 10,AT 922.5,
13140.    "F" 10,AT 925.5,"F" 10,AT 928.5,
13150.    "F" 10,AT 931.5,"F" 10,AT 934.5,
13160.    "F" 10,AT 937.5,"F" 10,AT 940.5,
13170.    "F" 10,AT 943.5,"F" 10,AT 946.5,
13180.    "F" 10,AT 949.5,"F" 10,AT 952.5,
13190.    "F" 10,AT 955.5,"F" 10,AT 958.5,
13200.    "F" 10,AT 961.5,"F" 10,AT 964.5,
13210.    "F" 10,AT 967.5,"F" 10,AT 970.5,
13220.    "F" 10,AT 973.5,"F" 10,AT 976.5,
13230.    "F" 10,AT 979.5,"F" 10,AT 982.5,
13240.    "F" 10,AT 985.5,"F" 10,AT 988.5,
13250.    "F" 10,AT 991.5,"F" 10,AT 994.5,
13260.    "F" 10,AT 997.5,"F" 10,AT 1000.5,
13270.    "F" 10,AT 1003.5,"F" 10,AT 1006.5,
13280.    "F" 10,AT 1009.5,"F" 10,AT 1012.5,
13290.    "F" 10,AT 1015.5,"F" 10,AT 1018.5,
13300.    "F" 10,AT 1021.5,"F" 10,AT 1024.5,
13310.    "F" 10,AT 1027.5,"F" 10,AT 1030.5,
13320.    "F" 10,AT 1033.5,"F" 10,AT 1036.5,
13330.    "F" 10,AT 1039.5,"F" 10,AT 1042.5,
13340.    "F" 10,AT 1045.5,"F" 10,AT 1048.5,
13350.    "F" 10,AT 1051.5,"F" 10,AT 1054.5,
13360.    "F" 10,AT 1057.5,"F" 10,AT 1060.5,
13370.    "F" 10,AT 1063.5,"F" 10,AT 1066.5,
13380.    "F" 10,AT 1069.5,"F" 10,AT 1072.5,
13390.    "F" 10,AT 1075.5,"F" 10,AT 1078.5,
13400.    "F" 10,AT 1081.5,"F" 10,AT 1084.5,
13410.    "F" 10,AT 1087.5,"F" 10,AT 1090.5,
13420.    "F" 10,AT 1093.5,"F" 10,AT 1096.5,
13430.    "F" 10,AT 1099.5,"F" 10,AT 1102.5,
13440.    "F" 10,AT 1105.5,"F" 10,AT 1108.5,
13450.    "F" 10,AT 1111.5,"F" 10,AT 1114.5,
13460.    "F" 10,AT 1117.5,"F" 10,AT 1120.5,
13470.    "F" 10,AT 1123.5,"F" 10,AT 1126.5,
13480.    "F" 10,AT 1129.5,"F" 10,AT 1132.5,
13490.    "F" 10,AT 1135.5,"F" 10,AT 1138.5,
13500.    "F" 10,AT 1141.5,"F" 10,AT 1144.5,
13510.    "F" 10,AT 1147.5,"F" 10,AT 1150.5,
13520.    "F" 10,AT 1153.5,"F" 10,AT 1156.5,
13530.    "F" 10,AT 1159.5,"F" 10,AT 1162.5,
13540.    "F" 10,AT 1165.5,"F" 10,AT 1168.5,
13550.    "F" 10,AT 1171.5,"F" 10,AT 1174.5,
13560.    "F" 10,AT 1177.5,"F" 10,AT 1180.5,
13570.    "F" 10,AT 1183.5,"F" 10,AT 1186.5,
13580.    "F" 10,AT 1189.5,"F" 10,AT 1192.5,
13590.    "F" 10,AT 1195.5,"F" 10,AT 1198.5,
13600.    "F" 10,AT 1201.5,"F" 10,AT 1204.5,
13610.    "F" 10,AT 1207.5,"F" 10,AT 1210.5,
13620.    "F" 10,AT 1213.5,"F" 10,AT 1216.5,
13630.    "F" 10,AT 1219.5,"F" 10,AT 1222.5,
13640.    "F" 10,AT 1225.5,"F" 10,AT 1228.5,
13650.    "F" 10,AT 1231.5,"F" 10,AT 1234.5,
13660.    "F" 10,AT 1237.5,"F" 10,AT 1240.5,
13670.    "F" 10,AT 1243.5,"F" 10,AT 1246.5,
13680.    "F" 10,AT 1249.5,"F" 10,AT 1252.5,
13690.    "F" 10,AT 1255.5,"F" 10,AT 1258.5,
13700.    "F" 10,AT 1261.5,"F" 10,AT 1264.5,
13710.    "F" 10,AT 1267.5,"F" 10,AT 1270.5,
13720.    "F" 10,AT 1273.5,"F" 10,AT 1276.5,
13730.    "F" 10,AT 1279.5,"F" 10,AT 1282.5,
13740.    "F" 10,AT 1285.5,"F" 10,AT 1288.5,
13750.    "F" 10,AT 1291.5,"F" 10,AT 1294.5,
13760.    "F" 10,AT 1297.5,"F" 10,AT 1300.5,
13770.    "F" 10,AT 1303.5,"F" 10,AT 1306.5,
13780.    "F" 10,AT 1309.5,"F" 10,AT 1312.5,
13790.    "F" 10,AT 1315.5,"F" 10,AT 1318.5,
13800.    "F" 10,AT 1321.5,"F" 10,AT 1324.5,
13810.    "F" 10,AT 1327.5,"F" 10,AT 1330.5,
13820.    "F" 10,AT 1333.5,"F" 10,AT 1336.5,
13830.    "F" 10,AT 1339.5,"F" 10,AT 1342.5,
13840.    "F" 10,AT 1345.5,"F" 10,AT 1348.5,
13850.    "F" 10,AT 1351.5,"F" 10,AT 1354.5,
13860.    "F" 10,AT 1357.5,"F" 10,AT 1360.5,
13870.    "F" 10,AT 1363.5,"F" 10,AT 1366.5,
13880.    "F" 10,AT 1369.5,"F" 10,AT 1372.5,
13890.    "F" 10,AT 1375.5,"F" 10,AT 1378.5,
13900.    "F" 10,AT 1381.5,"F" 10,AT 1384.5,
13910.    "F" 10,AT 1387.5,"F" 10,AT 1390.5,
13920.    "F" 10,AT 1393.5,"F" 10,AT 1396.5,
13930.    "F" 10,AT 1399.5,"F" 10,AT 1402.5,
13940.    "F" 10,AT 1405.5,"F" 10,AT 1408.5,
13950.    "F" 10,AT 1411.5,"F" 10,AT 1414.5,
13960.    "F" 10,AT 1417.5,"F" 10,AT 1420.5,
13970.    "F" 10,AT 1423.5,"F" 10,AT 1426.5,
13980.    "F" 10,AT 1429.5,"F" 10,AT 1432.5,
13990.    "F" 10,AT 1435.5,"F" 10,AT 1438.5,
14000.    "F" 10,AT 1441.5,"F" 10,AT 1444.5,
14010.    "F" 10,AT 1447.5,"F" 10,AT 1450.5,
14020.    "F" 10,AT 1453.5,"F" 10,AT 1456.5,
14030.    "F" 10,AT 1459.5,"F" 10,AT 1462.5,
14040.    "F" 10,AT 1465.5,"F" 10,AT 1468.5,
14050.    "F" 10,AT 1471.5,"F" 10,AT 1474.5,
14060.    "F" 10,AT 1477.5,"F" 10,AT 1480.5,
14070.    "F" 10,AT 1483.5,"F" 10,AT 1486.5,
14080.    "F" 10,AT 1489.5,"F" 10,AT 1492.5,
14090.    "F" 10,AT 1495.5,"F" 10,AT 1498.5,
14100.    "F" 10,AT 1501.5,"F" 10,AT 1504.5,
14110.    "F" 10,AT 1507.5,"F" 10,AT 1510.5,
14120.    "F" 10,AT 1513.5,"F" 10,AT 1516.5,
14130.    "F" 10,AT 1519.5,"F" 10,AT 1522.5,
14140.    "F" 10,AT 1525.5,"F" 10,AT 1528.5,
14150.    "F" 10,AT 1531.5,"F" 10,AT 1534.5,
14160.    "F" 10,AT 1537.5,"F" 10,AT 1540.5,
14170.    "F" 10,AT 1543.5,"F" 10,AT 1546.5,
14180.    "F" 10,AT 1549.5,"F" 10,AT 1552.5,
14190.    "F" 10,AT 1555.5,"F" 10,AT 1558.5,
14200.    "F" 10,AT 1561.5,"F" 10,AT 1564.5,
14210.    "F" 10,AT 1567.5,"F" 10,AT 1570.5,
14220.    "F" 10,AT 1573.5,"F" 10,AT 1576.5,
14230.    "F" 10,AT 1579.5,"F" 10,AT 1582.5,
14240.    "F" 10,AT 1585.5,"F" 10,AT 1588.5,
14250.    "F" 10,AT 1591.5,"F" 10,AT 1594.5,
14260.    "F" 10,AT 1597.5,"F" 10,AT 1600.5,
14270.    "F" 10,AT 1603.5,"F" 10,AT 1606.5,
14280.    "F" 10,AT 1609.5,"F" 10,AT 1612.5,
14290.    "F" 10,AT 1615.5,"F" 10,AT 1618.5,
14300.    "F" 10,AT 1621.5,"F" 10,AT 1624.5,
14310.    "F" 10,AT 1627.5,"F" 10,AT 1630.5,
14320.    "F" 10,AT 1633.5,"F" 10,AT 1636.5,
14330.    "F" 10,AT 1639.5,"F" 10,AT 1642.5,
14340.    "F" 10,AT 1645.5,"F" 10,AT 1648.5,
14350.    "F" 10,AT 1651.5,"F" 10,AT 1654.5,
14360.    "F" 10,AT 1657.5,"F" 10,AT 1660.5,
14370.    "F" 10,AT 1663.5,"F" 10,AT 1666.5,
14380.    "F" 10,AT 1669.5,"F" 10,AT 1672.5,
14390.    "F" 10,AT 1675.5,"F" 10,AT 1678.5,
14400.    "F" 10,AT 1681.5,"F" 10,AT 1684.5,
14410.    "F" 10,AT 1687.5,"F" 10,AT 1690.5,
14420.    "F" 10,AT 1693.5,"F" 10,AT 1696.5,
14430.    "F" 10,AT 1699.5,"F" 10,AT 1702.5,
14440.    "F" 10,AT 1705.5,"F" 10,AT 1708.5,
14450.    "F" 10,AT 1711.5,"F" 10,AT 1714.5,
14460.    "F" 10,AT 1717.5,"F" 10,AT 1720.5,
14470.    "F" 10,AT 1723.5,"F" 10,AT 1726.5,
14480.    "F" 10,AT 1729.5,"F" 10,AT 1732.5,
14490.    "F" 10,AT 1735.5,"F" 10,AT 1738.5,
14500.    "F" 10,AT 1741.5,"F" 10,AT 1744.5,
14510.    "F" 10,AT 1747.5,"F" 10,AT 1750.5,
14520.    "F" 10,AT 1753.5,"F" 10,AT 1756.5,
14530.    "F" 10,AT 1759.5,"F" 10,AT 1762.5,
14540.    "F" 10,AT 1765.5,"F" 10,AT 1768.5,
14550.    "F" 10,AT 1771.5,"F" 10,AT 1774.5,
14560.    "F" 10,AT 1777.5,"F" 10,AT 1780.5,
14570.    "F" 10,AT 1783.5,"F" 10,AT 1786.5,
14580.    "F" 10,AT 1789.5,"F" 10,AT 1792.5,
14590.    "F" 10,AT 1795.5,"F" 10,AT 1798.5,
14600.    "F" 10,AT 1801.5,"F" 10,AT 1804.5,
14610.    "F" 10,AT 1807.5,"F" 10,AT 1810.5,
14620.    "F" 10,AT 1813.5,"F" 10,AT 1816.5,
14630.    "F" 10,AT 1819.5,"F" 10,AT 1822.5,
14640.    "F" 10,AT 1825.5,"F" 10,AT 1828.5,
14650.    "F" 10,AT 1831.5,"F" 10,AT 1834.5,
14660.    "F" 10,AT 1837.5,"F" 10,AT 1840.5,
14670.    "F" 10,AT 1843.5,"F" 10,AT 1846.5,
14680.    "F" 10,AT 1849.5,"F" 10,AT 1852.5,
14690.    "F" 10,AT 1855.5,"F" 10,AT 1858.5,
14700.    "F" 10,AT 1861.5,"F" 10,AT 1864.5,
14710.    "F" 10,AT 1867.5,"F" 10,AT 1870.5,
14720.    "F" 10,AT 1873.5,"F" 10,AT 1876.5,
14730.    "F" 10,AT 1879.5,"F" 10,AT 1882.5,
14740.    "F" 10,AT 1885.5,"F" 10,AT 1888.5,
14750.    "F" 10,AT 1891.5,"F" 10,AT 1894.5,
14760.    "F" 10,AT 1897.5,"F" 10,AT 1900.5,
14770.    "F" 10,AT 1903.5,"F" 10,AT 
```


DEFENDING YOUR SPECTRUM

Can you defend the earth from the invading aliens?



Not an easy game to find! You have three ships, and your mission is to rid the planet's surface of invading aliens.

Each time you fire a missile at an invader and destroy it you will be awarded 60 points but watch out, your missiles may not always destroy the aliens. If an invader hits you then you will lose one of your ships. The only way to get more ships is to increase your score; you will get an extra ship when your score increases to 5,000, 10,000, 15,000 and 20,000.

The control keys for the game are as follows:

The 'I' key moves your ship down.

The 'J' key moves your ship up.

The 'F' key fires a missile.

The 'T' key stops the game.

Once the program has been typed in and RUN, you will have to wait four seconds for the game to start - not long really, when you consider the fate of Earth is in your hands!

```

0 BODDEN 0: PAPER 0: INK 7: C A: - SCORE 0
LS 00 SUB 1000
01 CLEO
02 LET A=0: LET K=0: LET D=0:
03 LET B=0: LET G=10: LET I=0
04 LET J=0
05 LET L=0
06 LET M=0
07 LET N=0
08 LET O=0
09 LET P=0
10 LET Q=0
11 LET R=0
12 LET S=0
13 LET T=0
14 LET U=0
15 LET V=0
16 LET W=0
17 LET X=0
18 LET Y=0
19 LET Z=0
20 LET AA=0
21 LET AB=0
22 LET AC=0
23 LET AD=0
24 LET AE=0
25 LET AF=0
26 LET AG=0
27 LET AH=0
28 LET AI=0
29 LET AJ=0
30 LET AK=0
31 LET AL=0
32 LET AM=0
33 LET AN=0
34 LET AO=0
35 LET AP=0
36 LET AQ=0
37 LET AR=0
38 LET AS=0
39 LET AT=0
40 LET AU=0
41 LET AV=0
42 LET AW=0
43 LET AX=0
44 LET AY=0
45 LET AZ=0
46 LET BA=0
47 LET BB=0
48 LET BC=0
49 LET BD=0
50 LET BE=0
51 LET BF=0
52 LET BG=0
53 LET BH=0
54 LET BI=0
55 LET BJ=0
56 LET BK=0
57 LET BL=0
58 LET BM=0
59 LET BN=0
60 LET BO=0
61 LET BP=0
62 LET BQ=0
63 LET BR=0
64 LET BS=0
65 LET BT=0
66 LET BU=0
67 LET BV=0
68 LET BW=0
69 LET BX=0
70 LET BY=0
71 LET BZ=0
72 LET CA=0
73 LET CB=0
74 LET CC=0
75 LET CD=0
76 LET CE=0
77 LET CF=0
78 LET CG=0
79 LET CH=0
80 LET CI=0
81 LET CJ=0
82 LET CK=0
83 LET CL=0
84 LET CM=0
85 LET CN=0
86 LET CO=0
87 LET CP=0
88 LET CQ=0
89 LET CR=0
90 LET CS=0
91 LET CT=0
92 LET CU=0
93 LET CV=0
94 LET CW=0
95 LET CX=0
96 LET CY=0
97 LET CZ=0
98 LET DA=0
99 LET DB=0
100 LET DC=0
101 LET DD=0
102 LET DE=0
103 LET DF=0
104 LET DG=0
105 LET DH=0
106 LET DI=0
107 LET DJ=0
108 LET DK=0
109 LET DL=0
110 LET DM=0
111 LET DN=0
112 LET DO=0
113 LET DP=0
114 LET DQ=0
115 LET DR=0
116 LET DS=0
117 LET DT=0
118 LET DU=0
119 LET DV=0
120 LET DW=0
121 LET DX=0
122 LET DY=0
123 LET DZ=0
124 LET EA=0
125 LET EB=0
126 LET EC=0
127 LET ED=0
128 LET EE=0
129 LET EF=0
130 LET EG=0
131 LET EH=0
132 LET EI=0
133 LET EJ=0
134 LET EK=0
135 LET EL=0
136 LET EM=0
137 LET EN=0
138 LET EO=0
139 LET EP=0
140 LET EQ=0
141 LET ER=0
142 LET ES=0
143 LET ET=0
144 LET EU=0
145 LET EV=0
146 LET EW=0
147 LET EX=0
148 LET EY=0
149 LET EZ=0
150 LET FA=0
151 LET FB=0
152 LET FC=0
153 LET FD=0
154 LET FE=0
155 LET FF=0
156 LET FG=0
157 LET FH=0
158 LET FI=0
159 LET FJ=0
160 LET FK=0
161 LET FL=0
162 LET FM=0
163 LET FN=0
164 LET FO=0
165 LET FP=0
166 LET FQ=0
167 LET FR=0
168 LET FS=0
169 LET FT=0
170 LET FU=0
171 LET FV=0
172 LET FW=0
173 LET FX=0
174 LET FY=0
175 LET FZ=0
176 LET GA=0
177 LET GB=0
178 LET GC=0
179 LET GD=0
180 LET GE=0
181 LET GF=0
182 LET GH=0
183 LET GI=0
184 LET GJ=0
185 LET GK=0
186 LET GL=0
187 LET GM=0
188 LET GN=0
189 LET GO=0
190 LET GP=0
191 LET GQ=0
192 LET GR=0
193 LET GS=0
194 LET GT=0
195 LET GU=0
196 LET GV=0
197 LET GW=0
198 LET GX=0
199 LET GY=0
200 LET GZ=0
201 LET HA=0
202 LET HB=0
203 LET HC=0
204 LET HD=0
205 LET HE=0
206 LET HF=0
207 LET HG=0
208 LET HH=0
209 LET HI=0
210 LET HJ=0
211 LET HK=0
212 LET HL=0
213 LET HM=0
214 LET HN=0
215 LET HO=0
216 LET HP=0
217 LET HQ=0
218 LET HR=0
219 LET HS=0
220 LET HT=0
221 LET HU=0
222 LET HV=0
223 LET HW=0
224 LET HX=0
225 LET HY=0
226 LET HZ=0
227 LET IA=0
228 LET IB=0
229 LET IC=0
230 LET ID=0
231 LET IE=0
232 LET IF=0
233 LET IG=0
234 LET IH=0
235 LET II=0
236 LET IJ=0
237 LET IK=0
238 LET IL=0
239 LET IM=0
240 LET IN=0
241 LET IO=0
242 LET IP=0
243 LET IQ=0
244 LET IR=0
245 LET IS=0
246 LET IT=0
247 LET IU=0
248 LET IV=0
249 LET IW=0
250 LET IX=0
251 LET IY=0
252 LET IZ=0
253 LET JA=0
254 LET JB=0
255 LET JC=0
256 LET JD=0
257 LET JE=0
258 LET JF=0
259 LET JG=0
260 LET JH=0
261 LET JI=0
262 LET JJ=0
263 LET JK=0
264 LET JL=0
265 LET JM=0
266 LET JN=0
267 LET JO=0
268 LET JP=0
269 LET JQ=0
270 LET JR=0
271 LET JS=0
272 LET JT=0
273 LET JU=0
274 LET JV=0
275 LET JW=0
276 LET JX=0
277 LET JY=0
278 LET JZ=0
279 LET KA=0
280 LET KB=0
281 LET KC=0
282 LET KD=0
283 LET KE=0
284 LET KF=0
285 LET KG=0
286 LET KH=0
287 LET KI=0
288 LET KJ=0
289 LET KK=0
290 LET KL=0
291 LET KM=0
292 LET KN=0
293 LET KO=0
294 LET KP=0
295 LET KQ=0
296 LET KR=0
297 LET KS=0
298 LET KT=0
299 LET KU=0
300 LET KV=0
301 LET KW=0
302 LET KX=0
303 LET KY=0
304 LET KZ=0
305 LET LA=0
306 LET LB=0
307 LET LC=0
308 LET LD=0
309 LET LE=0
310 LET LF=0
311 LET LG=0
312 LET LH=0
313 LET LI=0
314 LET LJ=0
315 LET LK=0
316 LET LL=0
317 LET LM=0
318 LET LN=0
319 LET LO=0
320 LET LP=0
321 LET LQ=0
322 LET LR=0
323 LET LS=0
324 LET LT=0
325 LET LU=0
326 LET LV=0
327 LET LW=0
328 LET LX=0
329 LET LY=0
330 LET LZ=0
331 LET MA=0
332 LET MB=0
333 LET MC=0
334 LET MD=0
335 LET ME=0
336 LET MF=0
337 LET MG=0
338 LET MH=0
339 LET MI=0
340 LET MJ=0
341 LET MK=0
342 LET ML=0
343 LET MM=0
344 LET MN=0
345 LET MO=0
346 LET MP=0
347 LET MQ=0
348 LET MR=0
349 LET MS=0
350 LET MT=0
351 LET MU=0
352 LET MV=0
353 LET MW=0
354 LET MX=0
355 LET MY=0
356 LET MZ=0
357 LET NA=0
358 LET NB=0
359 LET NC=0
360 LET ND=0
361 LET NE=0
362 LET NF=0
363 LET NG=0
364 LET NH=0
365 LET NI=0
366 LET NJ=0
367 LET NK=0
368 LET NL=0
369 LET NM=0
370 LET NN=0
371 LET NO=0
372 LET NP=0
373 LET NQ=0
374 LET NR=0
375 LET NS=0
376 LET NT=0
377 LET NU=0
378 LET NV=0
379 LET NW=0
380 LET NX=0
381 LET NY=0
382 LET NZ=0
383 LET OA=0
384 LET OB=0
385 LET OC=0
386 LET OD=0
387 LET OE=0
388 LET OF=0
389 LET OG=0
390 LET OH=0
391 LET OI=0
392 LET OJ=0
393 LET OK=0
394 LET OL=0
395 LET OM=0
396 LET ON=0
397 LET OO=0
398 LET OP=0
399 LET OQ=0
400 LET OR=0
401 LET OS=0
402 LET OT=0
403 LET OU=0
404 LET OV=0
405 LET OW=0
406 LET OX=0
407 LET OY=0
408 LET OZ=0
409 LET PA=0
410 LET PB=0
411 LET PC=0
412 LET PD=0
413 LET PE=0
414 LET PF=0
415 LET PG=0
416 LET PH=0
417 LET PI=0
418 LET PJ=0
419 LET PK=0
420 LET PL=0
421 LET PM=0
422 LET PN=0
423 LET PO=0
424 LET PP=0
425 LET PQ=0
426 LET PR=0
427 LET PS=0
428 LET PT=0
429 LET PU=0
430 LET PV=0
431 LET PW=0
432 LET PX=0
433 LET PY=0
434 LET PZ=0
435 LET QA=0
436 LET QB=0
437 LET QC=0
438 LET QD=0
439 LET QE=0
440 LET QF=0
441 LET QG=0
442 LET QH=0
443 LET QI=0
444 LET QJ=0
445 LET QK=0
446 LET QL=0
447 LET QM=0
448 LET QN=0
449 LET QO=0
450 LET QP=0
451 LET QQ=0
452 LET QR=0
453 LET QS=0
454 LET QT=0
455 LET QU=0
456 LET QV=0
457 LET QW=0
458 LET QX=0
459 LET QY=0
460 LET QZ=0
461 LET RA=0
462 LET RB=0
463 LET RC=0
464 LET RD=0
465 LET RE=0
466 LET RF=0
467 LET RG=0
468 LET RH=0
469 LET RI=0
470 LET RJ=0
471 LET RK=0
472 LET RL=0
473 LET RM=0
474 LET RN=0
475 LET RO=0
476 LET RP=0
477 LET RQ=0
478 LET RR=0
479 LET RS=0
480 LET RT=0
481 LET RU=0
482 LET RV=0
483 LET RW=0
484 LET RX=0
485 LET RY=0
486 LET RZ=0
487 LET SA=0
488 LET SB=0
489 LET SC=0
490 LET SD=0
491 LET SE=0
492 LET SF=0
493 LET SG=0
494 LET SH=0
495 LET SI=0
496 LET SJ=0
497 LET SK=0
498 LET SL=0
499 LET SM=0
500 LET SN=0
501 LET SO=0
502 LET SP=0
503 LET SQ=0
504 LET SR=0
505 LET SS=0
506 LET ST=0
507 LET SU=0
508 LET SV=0
509 LET SW=0
510 LET SX=0
511 LET SY=0
512 LET SZ=0
513 LET TA=0
514 LET TB=0
515 LET TC=0
516 LET TD=0
517 LET TE=0
518 LET TF=0
519 LET TG=0
520 LET TH=0
521 LET TI=0
522 LET TJ=0
523 LET TK=0
524 LET TL=0
525 LET TM=0
526 LET TN=0
527 LET TO=0
528 LET TP=0
529 LET TQ=0
530 LET TR=0
531 LET TS=0
532 LET TT=0
533 LET TU=0
534 LET TV=0
535 LET TW=0
536 LET TX=0
537 LET TY=0
538 LET TZ=0
539 LET UA=0
540 LET UB=0
541 LET UC=0
542 LET UD=0
543 LET UE=0
544 LET UF=0
545 LET UG=0
546 LET UH=0
547 LET UI=0
548 LET UJ=0
549 LET UK=0
550 LET UL=0
551 LET UM=0
552 LET UN=0
553 LET UO=0
554 LET UP=0
555 LET UQ=0
556 LET UR=0
557 LET US=0
558 LET UT=0
559 LET UU=0
560 LET UV=0
561 LET UW=0
562 LET UX=0
563 LET UY=0
564 LET UZ=0
565 LET VA=0
566 LET VB=0
567 LET VC=0
568 LET VD=0
569 LET VE=0
570 LET VF=0
571 LET VG=0
572 LET VH=0
573 LET VI=0
574 LET VJ=0
575 LET VK=0
576 LET VL=0
577 LET VM=0
578 LET VN=0
579 LET VO=0
580 LET VP=0
581 LET VQ=0
582 LET VR=0
583 LET VS=0
584 LET VT=0
585 LET VU=0
586 LET VV=0
587 LET VW=0
588 LET VX=0
589 LET VY=0
590 LET VZ=0
591 LET WA=0
592 LET WB=0
593 LET WC=0
594 LET WD=0
595 LET WE=0
596 LET WF=0
597 LET WG=0
598 LET WH=0
599 LET WI=0
600 LET WJ=0
601 LET WK=0
602 LET WL=0
603 LET WM=0
604 LET WN=0
605 LET WO=0
606 LET WP=0
607 LET WQ=0
608 LET WR=0
609 LET WS=0
610 LET WT=0
611 LET WU=0
612 LET WV=0
613 LET WW=0
614 LET WX=0
615 LET WY=0
616 LET WZ=0
617 LET XA=0
618 LET XB=0
619 LET XC=0
620 LET XD=0
621 LET XE=0
622 LET XF=0
623 LET XG=0
624 LET XH=0
625 LET XI=0
626 LET XJ=0
627 LET XK=0
628 LET XL=0
629 LET XM=0
630 LET XN=0
631 LET XO=0
632 LET XP=0
633 LET XQ=0
634 LET XR=0
635 LET XS=0
636 LET XT=0
637 LET XU=0
638 LET XV=0
639 LET XW=0
640 LET XX=0
641 LET XY=0
642 LET XZ=0
643 LET YA=0
644 LET YB=0
645 LET YC=0
646 LET YD=0
647 LET YE=0
648 LET YF=0
649 LET YG=0
650 LET YH=0
651 LET YI=0
652 LET YJ=0
653 LET YK=0
654 LET YL=0
655 LET YM=0
656 LET YN=0
657 LET YO=0
658 LET YP=0
659 LET YQ=0
660 LET YR=0
661 LET YS=0
662 LET YT=0
663 LET YU=0
664 LET YV=0
665 LET YW=0
666 LET YX=0
667 LET YY=0
668 LET YZ=0
669 LET ZA=0
670 LET ZB=0
671 LET ZC=0
672 LET ZD=0
673 LET ZE=0
674 LET ZF=0
675 LET ZG=0
676 LET ZH=0
677 LET ZI=0
678 LET ZJ=0
679 LET ZK=0
680 LET ZL=0
681 LET ZM=0
682 LET ZN=0
683 LET ZO=0
684 LET ZP=0
685 LET ZQ=0
686 LET ZR=0
687 LET ZS=0
688 LET ZT=0
689 LET ZU=0
690 LET ZV=0
691 LET ZW=0
692 LET ZX=0
693 LET ZY=0
694 LET ZZ=0
695 LET AA=0
696 LET AB=0
697 LET AC=0
698 LET AD=0
699 LET AE=0
700 LET AF=0
701 LET AG=0
702 LET AH=0
703 LET AI=0
704 LET AJ=0
705 LET AK=0
706 LET AL=0
707 LET AM=0
708 LET AN=0
709 LET AO=0
710 LET AP=0
711 LET AQ=0
712 LET AR=0
713 LET AS=0
714 LET AT=0
715 LET AU=0
716 LET AV=0
717 LET AW=0
718 LET AX=0
719 LET AY=0
720 LET AZ=0
721 LET BA=0
722 LET BB=0
723 LET BC=0
724 LET BD=0
725 LET BE=0
726 LET BF=0
727 LET BG=0
728 LET BH=0
729 LET BI=0
730 LET BJ=0
731 LET BK=0
732 LET BL=0
733 LET BM=0
734 LET BN=0
735 LET BO=0
736 LET BP=0
737 LET BQ=0
738 LET BR=0
739 LET BS=0
740 LET BT=0
741 LET BU=0
742 LET BV=0
743 LET BW=0
744 LET BX=0
745 LET BY=0
746 LET BZ=0
747 LET CA=0
748 LET CB=0
749 LET CC=0
750 LET CD=0
751 LET CE=0
752 LET CF=0
753 LET CG=0
754 LET CH=0
755 LET CI=0
756 LET CJ=0
757 LET CK=0
758 LET CL=0
759 LET CM=0
760 LET CN=0
761 LET CO=0
762 LET CP=0
763 LET CQ=0
764 LET CR=0
765 LET CS=0
766 LET CT=0
767 LET CU=0
768 LET CV=0
769 LET CW=0
770 LET CX=0
771 LET CY=0
772 LET CZ=0
773 LET DA=0
774 LET DB=0
775 LET DC=0
776 LET DD=0
777 LET DE=0
778 LET DF=0
779 LET DG=0
780 LET DH=0
781 LET DI=0
782 LET DJ=0
783 LET DK=0
784 LET DL=0
785 LET DM=0
786 LET DN=0
787 LET DO=0
788 LET DP=0
789 LET DQ=0
790 LET DR=0
791 LET DS=0
792 LET DT=0
793 LET DU=0
794 LET DV=0
795 LET DW=0
796 LET DX=0
797 LET DY=0
798 LET DZ=0
799 LET EA=0
800 LET EB=0
801 LET EC=0
802 LET ED=0
803 LET EE=0
804 LET EF=0
805 LET EG=0
806 LET EH=0
807 LET EI=0
808 LET EJ=0
809 LET EK=0
810 LET EL=0
811 LET EM=0
812 LET EN=0
813 LET EO=0
814 LET EP=0
815 LET EQ=0
816 LET ER=0
817 LET ES=0
818 LET ET=0
819 LET EU=0
820 LET EV=0
821 LET EW=0
822 LET EX=0
823 LET EY=0
824 LET EZ=0
825 LET FA=0
826 LET FB=0
827 LET FC=0
828 LET FD=0
829 LET FE=0
830 LET FF=0
831 LET FG=0
832 LET FH=0
833 LET FI=0
834 LET FJ=0
835 LET FK=0
836 LET FL=0
837 LET FM=0
838 LET FN=0
839 LET FO=0
840 LET FP=0
841 LET FQ=0
842 LET FR=0
843 LET FS=0
844 LET FT=0
845 LET FU=0
846 LET FV=0
847 LET FW=0
848 LET FX=0
849 LET FY=0
850 LET FZ=0
851 LET GA=0
852 LET GB=0
853 LET GC=0
854 LET GD=0
855 LET GE=0
856 LET GF=0
857 LET GH=0
858 LET GI=0
859 LET GJ=0
860 LET GK=0
861 LET GL=0
862 LET GM=0
863 LET GN=0
864 LET GO=0
865 LET GP=0
866 LET GQ=0
867 LET GR=0
868 LET GS=0
869 LET GT=0
870 LET GU=0
871 LET GV=0
872 LET GW=0
873 LET GX=0
874 LET GY=0
875 LET GZ=0
876 LET HA=0
877 LET HB=0
878 LET HC=0
879 LET HD=0
880 LET HE=0
881 LET HF=0
882 LET HG=0
883 LET HH=0
884 LET HI=0
885 LET HJ=0
886 LET HK=0
887 LET HL=0
888 LET HM=0
889 LET HN=0
890 LET HO=0
891 LET HP=0
892 LET HQ=0
893 LET HR=0
894 LET HS=0
895 LET HT=0
896 LET HU=0
897 LET HV=0
898 LET HW=0
899 LET HX=0
900 LET HY=0
901 LET HZ=0
902 LET IA=0
903 LET IB=0
904 LET IC=0
905 LET ID=0
906 LET IE=0
907 LET IF=0
908 LET IG=0
909 LET IH=0
910 LET II=0
911 LET IJ=0
912 LET IK=0
913 LET IL=0
914 LET IM=0
915 LET IN=0
916 LET IO=0
917 LET IP=0
918 LET IQ=0
919 LET IR=0
920 LET IS=0
921 LET IT=0
922 LET IU=0
923 LET IV=0
924 LET IW=0
925 LET IX=0
926 LET IY=0
927 LET IZ=0
928 LET JA=0
929 LET JB=0
930 LET JC=0
931 LET JD=0
932 LET JE=0
933 LET JF=0
934 LET JG=0
935 LET JH=0
936 LET JI=0
937 LET JJ=0
938 LET JK=0
939 LET JL=0
940 LET JM=0
941 LET JN=0
942 LET JO=0
943 LET JP=0
944 LET JQ=0
945 LET JR=0
946 LET JS=0
947 LET JT=0
948 LET JU=0
949 LET JV=0
950 LET JW=0
951 LET JX=0
952 LET JY=0
953 LET JZ=0
954 LET KA=0
955 LET KB=0
956 LET KC=0
957 LET KD=0
958 LET KE=0
959 LET KF=0
960 LET KG=0
961 LET KH=0
962 LET KI=0
963 LET KJ=0
964 LET KK=0
965 LET KL=0
966 LET KM=0
967 LET KN=0
968 LET KO=0
969 LET KP=0
970 LET KQ=0
971 LET KR=0
972 LET KS=0
973 LET KT=0
974 LET KU=0
975 LET KV=0
976 LET KW=0
977 LET KX=0
978 LET KY=0
979 LET KZ=0
980 LET LA=0
981 LET LB=0
982 LET LC=0
983 LET LD=0
984 LET LE=0
985 LET LF=0
986 LET LG=0
987 LET LH=0
988 LET LI=0
989 LET LJ=0
990 LET LK=0
991 LET LL=0
992 LET LM=0
993 LET LN=0
994 LET LO=0
995 LET LP=0
996 LET LQ=0
997 LET LR=0
998 LET LS=0
999 LET LT=0
1000 LET LU=0
1001 LET LV=0
1002 LET LW=0
1003 LET LX=0
1004 LET LY=0
1005 LET LZ=0
1006 LET MA=0
1007 LET MB=0
1008 LET MC=0
1009 LET MD=0
1010 LET ME=0
1011 LET MF=0
1012 LET MG=0
1013 LET MH=0
1014 LET MI=0
1015 LET MJ=0
1016 LET MK=0
1017 LET ML=0
1018 LET MM=0
1019 LET MN=0
1020 LET MO=0
1021 LET MP=0
1022 LET MQ=0
1023 LET MR=0
1024 LET MS=0
1025 LET MT=0
1026 LET MU=0
1027 LET MV=0
1028 LET MW=0
1029 LET MX=0
1030 LET MY=0
1031 LET MZ=0
1032 LET NA=0
1033 LET NB=0
1034 LET NC=0
1035 LET ND=0
1036 LET NE=0
1037 LET NF=0
1038 LET NG=0
1039 LET NH=0
1040 LET NI=0
1041 LET NJ=0
1042 LET NK=0
1043 LET NL=0
1044 LET NM=0
1045 LET NN=0
1046 LET NO=0
1047 LET NP=0
1048 LET NQ=0
1049 LET NR=0
1050 LET NS=0
1051 LET NT=0
1052 LET NU=0
1053 LET NV=0
1054 LET NW=0
1055 LET NX=0
1056 LET NY=0
1057 LET NZ=0
1058 LET OA=0
1059 LET OB=0
1060 LET OC=0
1061 LET OD=0
1062 LET OE=0
1063 LET OF=0
1064 LET OG=0
1065 LET OH=0
1066 LET OI=0
1067 LET OJ=0
1068 LET OK=0
1069 LET OL=0
1070 LET OM=0
1071 LET ON=0
1072 LET OO=0
1073 LET OP=0
1074 LET OQ=0
1075 LET OR=0
1076 LET OS=0
1077 LET OT=0
1078 LET OU=0
1079 LET OV=0
1080 LET OW=0
1081 LET OX=0
1082 LET OY=0
1083 LET OZ=0
1084 LET PA=0
1085 LET PB=0
1086 LET PC=0
1087 LET PD=0
1088 LET PE=0
1089 LET PF=0
1090 LET PG=0
1091 LET PH=0
1092 LET PI=0
1093 LET PJ=0
1094 LET PK=0
1095 LET PL=0
1096 LET PM=0
1097 LET PN=0
1098 LET PO=0
1099 LET PP=0
1100 LET PQ=0
1101 LET PR=0
1102 LET PS=0
1103 LET PT=0
1104 LET PU=0
1105 LET PV=0
1106 LET PW=0
1107 LET PX=0
1108 LET PY=0
1109 LET PZ=0
1110 LET QA=0
1111 LET QB=0
1112 LET QC=0
1113 LET QD=0
1114 LET QE=0
1115 LET QF=0
1116 LET QG=0
1117 LET QH=0
1118 LET QI=0
1119 LET QJ=0
1120 LET QK=0
1121 LET QL=0
1122 LET QM=0
1123 LET QN=0
1124 LET QO=0
1125 LET QP=0
1126 LET QQ=0
1127 LET QR=0
1128 LET QS=0
1129 LET QT=0
1130 LET QU=0
1131 LET QV=0
1132 LET QW=0
1133 LET QX=0
1134 LET QY=0
1135 LET QZ=0
1136 LET RA=0
1137 LET RB=0
1138 LET RC=0
1139 LET RD=0
1140 LET RE=0
1141 LET RF=0
1142 LET RG=0
1143 LET RH=0
1144 LET RI=0
1145 LET RJ=0
1146 LET RK=0
1147 LET RL=0
1148 LET RM=0
1149 LET RN=0
1150 LET RO=0
1151 LET RP=0
1152 LET RQ=0
1153 LET RR=0
1154 LET RS=0
1155 LET RT=0
1156 LET RU=0
1157 LET RV=0
1158 LET RW=0
1159 LET RX=0
1160 LET RY=0
1161 LET RZ=0
1162 LET SA=0
1163 LET SB=0
1164 LET SC=0
1165 LET SD=0
1166 LET SE=0
1167 LET SF=0
1168 LET SG=0
1169 LET SH=0
1170 LET SI=0
1171 LET SJ=0
1172 LET SK=0
1173 LET SL=0
1174 LET SM=0
1175 LET SN=0
1176 LET SO=0
1177 LET SP=0
1178 LET SQ=0
1179 LET SR=0
1180 LET SS=0
1181 LET ST=0
1182 LET SU=0
1183 LET SV=0
1184 LET SW=0
1185 LET SX=0
1186 LET SY=0
1187 LET SZ=0
1188 LET TA=0
1189 LET TB=0
1190 LET TC=0
1191 LET TD=0
1192 LET TE=0
1193 LET TF=0
1194 LET TG=0
1195 LET TH=0
1196 LET TI=0
1197 LET TJ=0
1198 LET TK=0
1199 LET TL=0
1200 LET TM=0
1201 LET TN=0
1202 LET TO=0
1203 LET TP=0
1204 LET TQ=0
1205 LET TR=0
1206 LET TS=0
1207 LET TT=0
1208 LET TU=0
1209 LET TV=0
1210 LET TW=0
1211 LET TX=0
1212 LET TY=0
1213 LET TZ=0
1214 LET UA=0
1215 LET UB=0
1216 LET UC=0
1217 LET UD=0
1218 LET UE=0
1219 LET UF=0
1220 LET UG=0
1221 LET UH=0
1222 LET UI=0
1223 LET UJ=0
1224 LET UK=0
1225 LET UL=0
1226 LET UM=0
1227 LET UN=0
1228 LET UO=0
1229 LET UP=0
1230 LET UQ=0
1231 LET UR=0
1232 LET US=0
1233 LET UT=0
1234 LET UU=0
1235 LET UV=0
1236 LET UW=0
1237 LET UX=0
1238 LET UY=0
1239 LET UZ=0
1240 LET VA=0
1241 LET VB=0
1242 LET VC=0
1243 LET VD=0
1244 LET VE=0
1245 LET VF=0
1246 LET VG=0
1247 LET VH=0
1248 LET VI=0
1249 LET VJ=0
1250 LET VK=0
1251 LET VL=0
1252 LET VM=0
1253 LET VN=0
1254 LET VO=0
1255 LET VP=0
1256 LET VQ=0
1257 LET VR=0
1258 LET VS=0
1259 LET VT=0
1260 LET VU=0
1261 LET VV=0
1262 LET VW=0
1263 LET VX=0
1264 LET VY=0
1265 LET VZ=0
1266 LET WA=0
1267 LET WB=0
1268 LET WC=0
1269 LET WD=0
1270 LET WE=0
1271 LET WF=0
1272 LET WG=0
1273 LET WH=0
1274 LET WI=0
1275 LET WJ=0
1276 LET WK=0
1277 LET WL=0
1278 LET WM=0
1279 LET WN=0
1280 LET WO=0
1281 LET WP=0
1282 LET WQ=0
1283 LET WR=0
1284 LET WS=0
1285 LET WT=0
1286 LET WU=0
1287 LET WV=0
1288 LET WW=0
1289 LET WX=0
1290 LET WY=0
1291 LET WZ=0
1292 LET XA=0
1293 LET XB=0
1294 LET XC=0
1295 LET XD=0
1296 LET XE=0
1297 LET XF=0
1298 LET XG=0
1299 LET XH=0
1300 LET XI=0
1301 LET XJ=0
1302 LET XK=0
1303 LET XL=0
1304 LET XM=0
1305 LET XN=0
1306 LET XO=0
1307 LET XP=0
1308 LET XQ=0
1309 LET XR=0
1310 LET XS=0
1311 LET XT=0
1312 LET XU=0
1313 LET XV=0
1314 LET XW=0
1315 LET XX=0
1316 LET XY=0
1317 LET XZ=0
1318 LET YA=0
1319 LET YB=0
1320 LET YC=0
1321 LET YD=0
1322 LET YE=0
1323 LET YF=0
1324 LET YG=0
1325 LET YH=0
1326 LET YI=0
1327 LET YJ=0
1328 LET YK=0
1329 LET YL=0
1330 LET YM=0
1331 LET YN=0
1332 LET YO=0
1333 LET YP=0
1334 LET YQ=0
1335 LET YR=0
1336 LET YS=0
1337 LET YT=0
1338 LET YU=0
1339 LET YV=0
1340 LET YW=0
1341 LET YX=0
1342 LET YY=0
1343 LET YZ=0
1344 LET ZA=0
1345 LET ZB=0
1346 LET ZC=0
1347 LET ZD=0
1348 LET ZE=0
1349 LET ZF=0
1350 LET ZG=0
1351 LET ZH=0
1352 LET ZI=0
1353 LET ZJ=0
1354 LET ZK=0
1355 LET ZL=0
1356 LET ZM=0
1357 LET ZN=0
1358 LET ZO=0
1359 LET ZP=0
1360 LET ZQ=0
1361 LET ZR=0
1362 LET ZS=0
1363 LET ZT=0
1364 LET ZU=0
1365 LET ZV=0
1366 LET ZW=0
1367 LET ZX=0
1368 LET ZY=0
1369 LET ZZ=0
1370 LET AA=0
1371 LET AB=0
1372 LET AC=0
1373 LET AD=0
1374 LET AE=0
1375 LET AF=0
1376 LET AG=0
1377 LET AH=0
1378 LET AI=0
1379
```


SPECTRAMON

Find out how your Spectrum works with this excellent program.



If you've ever wondered how your ZX Spectrum works, Spectramon (the Spectrum monitor) will make it easy for you to find out. This program will print or display the contents of ROM or RAM in numeric, character or assembly language form. Addresses may be entered in decimal or hexadecimal, and the user may select the base used for output.

Spectramon will run on a 48K Spectrum with or without a printer. The disassembler option has been written with the listings of Z80 and Z81 programs in mind — unlike other published listings it will handle all 654 standard Z80 instruction codes, using the standard mnemonic names and formats devised by Zilog, the firm which designed the Z80 processor used in the Spectrum.

most complicated of any 8-bit microcomputer. The Z80 processor was designed by a group of people who left Intel, the firm which makes the 8080 processor, to set up Zilog. The Z80 will execute any of the instructions of an 8080, plus a large number of extra ones 'tacked on' by Zilog. This approach meant that programs written to run on an 8080 would also run on a Z80 without changes. New programs could then be written using the added facilities of the Z80. That was how many early Z80 programs were produced. The BASIC interpreter used on the TRS-80, for instance, is substantially an 8080 program even though the TRS-80 has a Z80 processor. Only the display and keyboard routines contain Z80 instructions since they were the last to be written.

Standard BASIC is written using the full features of the Z80 processor. Zilog added instructions to handle bit

moving and searching of tables in memory, extra registers (internal storage) and instructions to increase the number of things that could be done with the original 8080 registers. They wanted to more than double the number of possible instructions, but there was a problem — Intel had decided to use a single byte (8 bits) to store the instruction numbers for the 8080, and most of the 256 possible numbers were already in use.

Zilog got around this by giving four instruction numbers special meanings — instructions with one of these numbers would carry out a certain 'class' of operation, and the next byte would explain the operation required in detail. In theory, that gave Zilog plenty of possible numbers — 252 (using the remaining one-byte values) plus 1,024 (4×256) if they were to use all of the possible two-byte instructions. In practice, they only used 694 of the 1,276 possibilities, but that's still a very large number of instructions for an 8-bit computer!

If you consult Appendix A of a Spectrum manual, you will see the standard Z80 mnemonics listed. The prefix byte 200 is used to generate add-on instructions for bitwise operations — instructions which manipulate or test binary digits. The prefix byte, 221, indicates that the next instruction is an 8080 one, which would use register pair HL but must now use register IX instead. Likewise, the prefix, 231, indicates that IX should replace HL in the next instruction. If HL was in brackets in the old 8080 instruction, as in LD B, (HL), then the Z80 version allows an effect to be applied to IX or IF before use — this is specified in an extra byte after the end of the 8080 instruction. Finally, the prefix 237 is used to indicate that the instruction following is

Z80 INSTRUCTIONS

The Z80 instruction set is the

one of a group of miscellaneous Z80 add-ons.

MONITORING THE SITUATION

If all this sounds very complicated you've probably realized why a monitor is a useful program — Spectramon will automatically convert sequences like 'EDH 7BH 3DH 9CH' into the mnemonic, LD SP, (23613). The EDH told Spectramon that it was a miscellaneous Z80 add-on instruction. (EDH is 230 decimal). The 7BH corresponds to LD SP, (some address), and the 3DH 9CH corresponds to the value 23613. To check that, convert 3DH and 9CH to decimal, then add the first result to the second (multiplied by 256). It's an awful lot simpler to let the computer puzzle that out than it is to work it out for every instruction by hand.

Of course, you may think that LD SP, (23613), is just as baffling as EDH 7BH 3DH 9CH — in which case, you'll have to learn a little about Z80 machine code before Spectramon becomes useful to you. Before you can investigate the ROM of a computer, you do need to understand the computer language in which it was written — assembler. In most cases, LD SP, (23613) is an assembler (or 'assembly language' or 'machine code') command. If you don't understand assembler, please don't throw this article away! It will take you no longer to learn assembler than it did to learn BASIC! (It should be just as much fun too) and you can come back to Spectramon when you know more.

In fact, the instruction, LD SP, (23613) has a very simple purpose — it tells the computer to put the number in address 23613 into the register called 'SP'. If you consult the Spectrum manual you will find out that 23613 is the 'address of item on machine stack to be used as error return', which tells you that the instruction is part of the ROM error-handler. Using a disassembled listing and the table of 'System

variables' in chapter 25 of the manual, you can trace your way through the ROM, finding out what each section does.

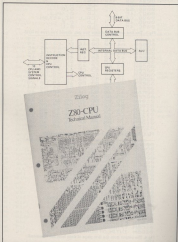
USING THE PROGRAM

Spectramon takes about 15 seconds to set itself up when first RUN. During this time, it is building a table of instruction codes for the disassembler, and once that is complete, the menu of commands will appear. Figure 1 shows this display.

To quit from the monitor, type 'Q' followed by Enter. This returns you to ZX BASIC. If you wish to disassemble a program in RAM or ROM, then you should type 'D' followed immediately by the address at which you want to start. Addresses may be entered to

Spectramon in decimal or Hex — if you want to disassemble from address 128 (decimal), you could type D 128 or D000EH or D7EH — leading zeros are optional — and if you enter more than four Hex digits, only the last four will be considered. If a meaningless address is typed (such as D, DFF, D123456 or D-D) then the command will be ignored.

The disassembler displays the contents of memory one screenful (21 lines) at a time. Figure 2 shows a disassembly of the first screenful of the Spectrum ROM. The left-hand column shows the address of the instruction. It is followed on the same line by a hexadecimal representation of the instruction, and then the



0 044 036 0

RECEIVED FOR THE DIRECTOR

Business Development Process

Symposium: Public Health Research

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 365–372

[illegible]

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

Fig. 1. The workday schedule offered for New Mexicans.

```

0007E 210BFE LD HL,FED0H
00081 80 DEC C
00082 06 SET C
00083 FE10 LD HL,10H
00086 06 SET C
00088 FE18 LD HL,18H
0008B 3F CDF
0008E 08 RET
00091 33 INC HL
00093 FE10 LD HL,10H
00096 30001 JH C,0000H
0009F 33 INC HL
000A0 37 SCF
000A1 22505C LD HL,10050H),HL
000A4 C9 RET
000A5 0F LD B
000A8 33 INC B
000AB 4F LD C,HL)
000AE 04494E LD HL,4E49H
000B0 48 LD C,E
000B3 48 LD B,L

```

Fig. 1. The first two modes of vibration of the cylindrical shell type in [2].

[illegible]

Fig. 4. The first expanded column in the column from 11.13.

0000H	END INKEY SP PHF01
0001H	END SCREENBOTTLE
0002H	HEPUL BOCCEVALLEN
0003H	3 INCOOTRANIGLES
0004H	THE NEXT IN SCREEN
0005H	AND PRA INUSPSTR
0006H	CH ANDNOTHOMAND
0100H	111111 INETHOMAND
0110H	TOPPER PNCOTROH
0120H	ATCOFFERBACROH
0130H	BOLOS ENOCEPUL
0140H	LYNDEPCIRCLEINS
0150H	COFFER GASHIGHT
0160H	INVERSECOUL
0170H	MINILLISTOPPER
0180H	COFFERSTOMPER
0190H	COFFERSTOMPER
0200H	COFFERSTOMPER
0210H	COFFERSTOMPER
0220H	COFFERSTOMPER
0230H	COFFERSTOMPER
0240H	COFFERSTOMPER
0250H	COFFERSTOMPER

Fig. 4. The output from the first page of data from the simulated RDS.

assembly language test. After 21 lines have been displayed, the prompt More? (Enter - No) will appear. Press any alphabetic or numeric key and the listing will continue on a new screen. Press the Enter key to return to the menu.

After each line is displayed, the program checks to see whether or not a key has been pressed. The Space key pauses the display, which will continue when any alphanumeric key is pressed. The Enter key causes disassembly to cease and the screen is displayed.

MAGIC NUMBER?

The third option allows display of the numeric contents of memory. Although the disassembler does this, it only lists between one and four bytes per line (depending upon the instruction). The **N** command allows eight bytes to be listed on each line of the display. A start address may be specified in Hex or decimal, just as for the **D** command.

The **N** command is useful for displaying the contents of tables used by a program or the ROM. Type **N150** to see the Spectrum reserved-word table. That is where Z8 BASIC stores the spellings of words such as **PRINT** and **RETURN**. The words are stored in a modified version of ASCII code — the last letter of each word has 128 (80H) added to it, to make it easy for the ROM routine which displays words to find where each one starts and ends.

If you found the numeric representation of the BASIC words rather hard to follow, you can use the command `ASCII` to display the reserved word table in character form. The command uses 3-bit ASCII values, so that letters with 128th bits added to their code will print out correctly. To avoid changing columns or moving the cursor unexpectedly try, `ASCII` output routine displays control characters (those with a code less than 32) as full stops.

You can use the Space and Enter keys to control listings output by commands `M` and `A`, just as you would for a disassembly. Even `U` lines the

"More?" (Enter = No) message will appear before a new screen is started.

The final two commands themselves, but they do change the output which the others generate. When you first RUN Spectramon, the message "P Printer option (Now ON)" appears. Type the command "P" followed by Enter and the message will become "P Printer option (Now OFF)". If you then display memory contents (using A, M or D) the information will be sent to the printer as well as the television. Once you've finished printing, press Enter to stop the display and then use the command "P" to switch the printer option off again.

Notice that the printer routine does not output any lines until an entire screen-full has been generated. In fact, it deliberately avoids using the LPRINT statement to send each line to the printer. Instead it uses COPY, the ZX BASIC command which sends all of the text on the screen to the printer. That's because it is almost twice as fast to build up a full screenfull of data and then print it using COPY than it is to use LPRINT for each line as it is generated.

The printer can't stop and start very quickly and consequently the LPRINT statement is much slower than COPY — the printer must rev up and slow down 21 times (once for each line) instead of just once. In fact, the printer always outputs the last line of a group of half speed, to make sure that everything falls in the correct place when it stops. As far as it is concerned each LPRINT is the last line of a group (when there's less than 33 characters being printed).

BASE CHOICE

The final option allows the user to select the base in which numbers are output by the program. Sometimes it is useful to have numbers printed in decimal (for example, when referring to addresses mentioned in the Spectrum manual) and sometimes hexadecimal is more convenient (when displaying

File	Z80 Assembler	— after ON	— after OFF	File	Z80 Assembler	— after ON	— after OFF
01	map	no o		44	no o	no o	no o
02	no b	no o		45	no o	no o	no o
03	no b	no o		46	no o	no o	no o
04	no b	no o		47	no o	no o	no o
05	no b	no o		48	no o	no o	no o
06	no b	no o		49	no o	no o	no o
07	no b	no o		50	no o	no o	no o
08	no b	no o		51	no o	no o	no o
09	no b	no o		52	no o	no o	no o
10	no b	no o		53	no o	no o	no o
11	no b	no o		54	no o	no o	no o
12	no b	no o		55	no o	no o	no o
13	no b	no o		56	no o	no o	no o
14	no b	no o		57	no o	no o	no o
15	no b	no o		58	no o	no o	no o
16	no b	no o		59	no o	no o	no o
17	no b	no o		60	no o	no o	no o
18	no b	no o		61	no o	no o	no o
19	no b	no o		62	no o	no o	no o
20	no b	no o		63	no o	no o	no o
21	no b	no o		64	no o	no o	no o
22	no b	no o		65	no o	no o	no o
23	no b	no o		66	no o	no o	no o
24	no b	no o		67	no o	no o	no o
25	no b	no o		68	no o	no o	no o
26	no b	no o		69	no o	no o	no o
27	no b	no o		70	no o	no o	no o
28	no b	no o		71	no o	no o	no o
29	no b	no o		72	no o	no o	no o
30	no b	no o		73	no o	no o	no o
31	no b	no o		74	no o	no o	no o
32	no b	no o		75	no o	no o	no o
33	no b	no o		76	no o	no o	no o
34	no b	no o		77	no o	no o	no o
35	no b	no o		78	no o	no o	no o
36	no b	no o		79	no o	no o	no o
37	no b	no o		80	no o	no o	no o
38	no b	no o		81	no o	no o	no o
39	no b	no o		82	no o	no o	no o
40	no b	no o		83	no o	no o	no o
41	no b	no o		84	no o	no o	no o
42	no b	no o		85	no o	no o	no o
43	no b	no o		86	no o	no o	no o
44	no b	no o		87	no o	no o	no o
45	no b	no o		88	no o	no o	no o
46	no b	no o		89	no o	no o	no o
47	no b	no o		90	no o	no o	no o
48	no b	no o		91	no o	no o	no o
49	no b	no o		92	no o	no o	no o
50	no b	no o		93	no o	no o	no o
51	no b	no o		94	no o	no o	no o
52	no b	no o		95	no o	no o	no o
53	no b	no o		96	no o	no o	no o
54	no b	no o		97	no o	no o	no o
55	no b	no o		98	no o	no o	no o
56	no b	no o		99	no o	no o	no o
57	no b	no o		100	no o	no o	no o
58	no b	no o		101	no o	no o	no o
59	no b	no o		102	no o	no o	no o
60	no b	no o		103	no o	no o	no o
61	no b	no o		104	no o	no o	no o
62	no b	no o		105	no o	no o	no o
63	no b	no o		106	no o	no o	no o
64	no b	no o		107	no o	no o	no o
65	no b	no o		108	no o	no o	no o
66	no b	no o		109	no o	no o	no o
67	no b	no o		110	no o	no o	no o
68	no b	no o		111	no o	no o	no o
69	no b	no o		112	no o	no o	no o
70	no b	no o		113	no o	no o	no o
71	no b	no o		114	no o	no o	no o
72	no b	no o		115	no o	no o	no o
73	no b	no o		116	no o	no o	no o
74	no b	no o		117	no o	no o	no o
75	no b	no o		118	no o	no o	no o
76	no b	no o		119	no o	no o	no o
77	no b	no o		120	no o	no o	no o
78	no b	no o		121	no o	no o	no o
79	no b	no o		122	no o	no o	no o
80	no b	no o		123	no o	no o	no o
81	no b	no o		124	no o	no o	no o
82	no b	no o		125	no o	no o	no o
83	no b	no o		126	no o	no o	no o
84	no b	no o		127	no o	no o	no o
85	no b	no o		128	no o	no o	no o
86	no b	no o		129	no o	no o	no o
87	no b	no o		130	no o	no o	no o
88	no b	no o		131	no o	no o	no o
89	no b	no o		132	no o	no o	no o
90	no b	no o		133	no o	no o	no o
91	no b	no o		134	no o	no o	no o
92	no b	no o		135	no o	no o	no o
93	no b	no o		136	no o	no o	no o
94	no b	no o		137	no o	no o	no o
95	no b	no o		138	no o	no o	no o
96	no b	no o		139	no o	no o	no o
97	no b	no o		140	no o	no o	no o
98	no b	no o		141	no o	no o	no o
99	no b	no o		142	no o	no o	no o
100	no b	no o		143	no o	no o	no o
101	no b	no o		144	no o	no o	no o
102	no b	no o		145	no o	no o	no o
103	no b	no o		146	no o	no o	no o
104	no b	no o		147	no o	no o	no o
105	no b	no o		148	no o	no o	no o
106	no b	no o		149	no o	no o	no o
107	no b	no o		150	no o	no o	no o
108	no b	no o		151	no o	no o	no o
109	no b	no o		152	no o	no o	no o
110	no b	no o		153	no o	no o	no o
111	no b	no o		154	no o	no o	no o
112	no b	no o		155	no o	no o	no o
113	no b	no o		156	no o	no o	no o
114	no b	no o		157	no o	no o	no o
115	no b	no o		158	no o	no o	no o
116	no b	no o		159	no o	no o	no o
117	no b	no o		160	no o	no o	no o
118	no b	no o		161	no o	no o	no o
119	no b	no o		162	no o	no o	no o
120	no b	no o		163	no o	no o	no o
121	no b	no o		164	no o	no o	no o
122	no b	no o		165	no o	no o	no o
123	no b	no o		166	no o	no o	no o
124	no b	no o		167	no o	no o	no o
125	no b	no o		168	no o	no o	no o
126	no b	no o		169	no o	no o	no o
127	no b	no o		170	no o	no o	no o
128	no b	no o		171	no o	no o	no o
129	no b	no o		172	no o	no o	no o
130	no b	no o		173	no o	no o	no o
131	no b	no o		174	no o	no o	no o
132	no b	no o		175	no o	no o	no o
133	no b	no o		176	no o	no o	no o
134	no b	no o		177	no o	no o	no o
135	no b	no o		178	no o	no o	no o
136	no b	no o		179	no o	no o	no o
137	no b	no o		180	no o	no o	no o
138	no b	no o		181	no o	no o	no o
139	no b	no o		182	no o	no o	no o
140	no b	no o		183	no o	no o	no o
141	no b	no o		184	no o	no o	no o
142	no b	no o		185	no o	no o	no o
143	no b	no o		186	no o	no o	no o
144	no b	no o		187	no o	no o	no o
145	no b	no o		188	no o	no o	no o
146	no b	no o		189	no o	no o	no o
147	no b	no o		190	no o	no o	no o
148	no b	no o		191	no o	no o	no o
149	no b	no o		192	no o	no o	no o
150	no b	no o		193	no o	no o	no o
151	no b	no o		194	no o	no o	no o
152	no b	no o		195	no o	no o	no o
153	no b	no o		196	no o	no o	no o
154	no b	no o		197	no o	no o	no o
155	no b	no o		198	no o	no o	no o
156	no b	no o		199	no o	no o	no o
157	no b	no o		200	no o	no o	no o
158	no b	no o		201	no o	no o	no o
159	no b	no o		202	no o	no o	no o
160	no b	no o		203	no o	no o	no o
161	no b	no o		204	no o	no o	no o
162	no b	no o		205	no o	no o	no o
163	no b	no o		206	no o	no o	no o
164	no b	no o		207	no o	no o	no o
165	no b	no o		208	no o	no o	no o
166	no b	no o		209	no o	no o	no o
167	no b	no o		210	no o	no o	no o
168	no b	no o		211	no o	no o	no o
169	no b	no o		212	no o	no o	no o
170	no b	no o		213	no o	no o	no o
171	no b	no o		214	no o	no o	no o
172	no b	no o		215	no o	no o	no o
173	no b	no o		216	no o	no o	no o
174	no b	no o		217	no o	no o	no o
175	no b	no o		218	no o	no o	no o
176	no b	no o		219	no o	no o	no o
177	no b	no o		220	no o	no o	no o
178	no b	no o		221	no o	no o	no o
179	no b	no o		222	no o	no o	no o
180	no b	no o		223	no o	no o	no o
181	no b	no o		224	no o	no o	no o
182	no b	no o		225	no o	no o	no o
183	no b	no o		226	no o	no o	no o
184	no b	no o		227	no o	no o	no o
185	no b	no o		228	no o	no o	no o
186	no b	no o		229	no o	no o	no o
187	no b	no o		230	no o	no o	no o
188	no b	no o		231	no o	no o	no o
189	no b	no o		232	no o	no o	no o
190	no b	no o		233	no o	no o	no o
191	no b	no o		234	no o	no o	no o
192	no b	no o		235	no o	no o	no o
193	no b	no o		236	no o	no o	no o
194	no b	no o		237	no o	no o	no o
195	no b	no o		238	no o	no o	no o
196	no b	no o		239	no o	no o	no o
197	no b	no o		240	no o	no o	no o
198	no b	no o		241	no o	no o	no o
199	no b	no o		242	no o	no o	no o
200	no b	no o		243	no o	no o	no o
201	no b	no o		244	no o	no o	no o
202	no b	no o		245	no o	no o	no o
203	no b	no o		246	no o	no o	no o
204	no b	no o		247	no o	no o	no o
205	no b	no o		248	no o	no o	no o
206	no b	no o		249	no o	no o	no o

Hex	Z80 Assembly	— after C8	— after D2	Hex	Z80 Assembly	— after C8	— after D2
00	sub h	hex 2.0		08	set bc	set 2.0	
01	sub l	hex 2.1		09	set de	set 2.1	
02	sub hl	hex 2.00		0A	set hl	set 2.0	
03	sub a	hex 2.0		0B	set hl	set 2.0	
04	stc 2.0	hex 2.0		0C	set hl	set 2.0	
05	stc 2.1	hex 2.1		0D	push bc	set 2.1	
06	stc 2.0	hex 2.0		0E	sub h	hex 2.00	
07	stc 2.0	hex 2.0		0F	sub l	hex 2.0	
08	stc 2.0	hex 2.0		10	sub a	hex 2.0	
09	stc 2.0	hex 2.0		11	sub	hex 2.0	
0A	stc 2.0	hex 2.0		12	set 2.00	hex 2.0	
0B	stc 2.0	hex 2.0		13	set 2.00	hex 2.0	
0C	stc 2.0	hex 2.0		14	set 2.00	hex 2.0	
0D	stc 2.0	hex 2.0		15	set 2.00	hex 2.0	
0E	stc 2.0	hex 2.0		16	set 2.00	hex 2.0	
0F	stc 2.0	hex 2.0		17	set 2.00	hex 2.0	
10	and h	hex 2.0		18	set 2.00	hex 2.0	
11	and l	hex 2.1		19	set 2.00	hex 2.0	
12	and a	hex 2.0		1A	set 2.00	hex 2.0	
13	and h	hex 2.0		1B	set 2.00	hex 2.0	
14	and l	hex 2.1		1C	set 2.00	hex 2.0	
15	and a	hex 2.0		1D	set 2.00	hex 2.0	
16	and hl	hex 2.00		1E	set 2.00	hex 2.0	
17	and a	hex 2.0		1F	set 2.00	hex 2.0	
18	set h	hex 2.0		20	set 2.00	hex 2.0	
19	set l	hex 2.1		21	set 2.00	hex 2.0	
1A	set a	hex 2.0		22	set 2.00	hex 2.0	
1B	set h	hex 2.0		23	set 2.00	hex 2.0	
1C	set l	hex 2.1		24	set 2.00	hex 2.0	
1D	set a	hex 2.0		25	set 2.00	hex 2.0	
1E	set h	hex 2.0		26	set 2.00	hex 2.0	
1F	set l	hex 2.1		27	set 2.00	hex 2.0	
20	or h	hex 2.0		28	set 2.00	hex 2.0	
21	or l	hex 2.1		29	set 2.00	hex 2.0	
22	or a	hex 2.0		2A	set 2.00	hex 2.0	
23	or h	hex 2.0		2B	set 2.00	hex 2.0	
24	or l	hex 2.1		2C	set 2.00	hex 2.0	
25	or a	hex 2.0		2D	set 2.00	hex 2.0	
26	or hl	hex 2.00		2E	set 2.00	hex 2.0	
27	or a	hex 2.0		2F	set 2.00	hex 2.0	
28	or h	hex 2.0		30	set 2.00	hex 2.0	
29	or l	hex 2.1		31	set 2.00	hex 2.0	
2A	or a	hex 2.0		32	set 2.00	hex 2.0	
2B	or h	hex 2.0		33	set 2.00	hex 2.0	
2C	or l	hex 2.1		34	set 2.00	hex 2.0	
2D	or a	hex 2.0		35	set 2.00	hex 2.0	
2E	or hl	hex 2.00		36	set 2.00	hex 2.0	
2F	or a	hex 2.0		37	set 2.00	hex 2.0	
30	or h	hex 2.0		38	set 2.00	hex 2.0	
31	or l	hex 2.1		39	set 2.00	hex 2.0	
32	or a	hex 2.0		3A	set 2.00	hex 2.0	
33	or h	hex 2.0		3B	set 2.00	hex 2.0	
34	or l	hex 2.1		3C	set 2.00	hex 2.0	
35	or a	hex 2.0		3D	set 2.00	hex 2.0	
36	or hl	hex 2.00		3E	set 2.00	hex 2.0	
37	or a	hex 2.0		3F	set 2.00	hex 2.0	
38	or h	hex 2.0		40	set 2.00	hex 2.0	
39	or l	hex 2.1		41	set 2.00	hex 2.0	
3A	or a	hex 2.0		42	set 2.00	hex 2.0	
3B	or h	hex 2.0		43	set 2.00	hex 2.0	
3C	or l	hex 2.1		44	set 2.00	hex 2.0	
3D	or a	hex 2.0		45	set 2.00	hex 2.0	
3E	or hl	hex 2.00		46	set 2.00	hex 2.0	
3F	or a	hex 2.0		47	set 2.00	hex 2.0	
40	or h	hex 2.0		48	set 2.00	hex 2.0	
41	or l	hex 2.1		49	set 2.00	hex 2.0	
42	or a	hex 2.0		4A	set 2.00	hex 2.0	
43	or h	hex 2.0		4B	set 2.00	hex 2.0	
44	or l	hex 2.1		4C	set 2.00	hex 2.0	
45	or a	hex 2.0		4D	set 2.00	hex 2.0	
46	or hl	hex 2.00		4E	set 2.00	hex 2.0	
47	or a	hex 2.0		4F	set 2.00	hex 2.0	
48	or h	hex 2.0		50	set 2.00	hex 2.0	
49	or l	hex 2.1		51	set 2.00	hex 2.0	
4A	or a	hex 2.0		52	set 2.00	hex 2.0	
4B	or h	hex 2.0		53	set 2.00	hex 2.0	
4C	or l	hex 2.1		54	set 2.00	hex 2.0	
4D	or a	hex 2.0		55	set 2.00	hex 2.0	
4E	or hl	hex 2.00		56	set 2.00	hex 2.0	
4F	or a	hex 2.0		57	set 2.00	hex 2.0	
50	or h	hex 2.0		58	set 2.00	hex 2.0	
51	or l	hex 2.1		59	set 2.00	hex 2.0	
52	or a	hex 2.0		5A	set 2.00	hex 2.0	
53	or h	hex 2.0		5B	set 2.00	hex 2.0	
54	or l	hex 2.1		5C	set 2.00	hex 2.0	
55	or a	hex 2.0		5D	set 2.00	hex 2.0	
56	or hl	hex 2.00		5E	set 2.00	hex 2.0	
57	or a	hex 2.0		5F	set 2.00	hex 2.0	
58	or h	hex 2.0		60	set 2.00	hex 2.0	
59	or l	hex 2.1		61	set 2.00	hex 2.0	
5A	or a	hex 2.0		62	set 2.00	hex 2.0	
5B	or h	hex 2.0		63	set 2.00	hex 2.0	
5C	or l	hex 2.1		64	set 2.00	hex 2.0	
5D	or a	hex 2.0		65	set 2.00	hex 2.0	
5E	or hl	hex 2.00		66	set 2.00	hex 2.0	
5F	or a	hex 2.0		67	set 2.00	hex 2.0	
60	or h	hex 2.0		68	set 2.00	hex 2.0	
61	or l	hex 2.1		69	set 2.00	hex 2.0	
62	or a	hex 2.0		6A	set 2.00	hex 2.0	
63	or h	hex 2.0		6B	set 2.00	hex 2.0	
64	or l	hex 2.1		6C	set 2.00	hex 2.0	
65	or a	hex 2.0		6D	set 2.00	hex 2.0	
66	or hl	hex 2.00		6E	set 2.00	hex 2.0	
67	or a	hex 2.0		6F	set 2.00	hex 2.0	
68	or h	hex 2.0		70	set 2.00	hex 2.0	
69	or l	hex 2.1		71	set 2.00	hex 2.0	
6A	or a	hex 2.0		72	set 2.00	hex 2.0	
6B	or h	hex 2.0		73	set 2.00	hex 2.0	
6C	or l	hex 2.1		74	set 2.00	hex 2.0	
6D	or a	hex 2.0		75	set 2.00	hex 2.0	
6E	or hl	hex 2.00		76	set 2.00	hex 2.0	
6F	or a	hex 2.0		77	set 2.00	hex 2.0	
70	or h	hex 2.0		78	set 2.00	hex 2.0	
71	or l	hex 2.1		79	set 2.00	hex 2.0	
72	or a	hex 2.0		7A	set 2.00	hex 2.0	
73	or h	hex 2.0		7B	set 2.00	hex 2.0	
74	or l	hex 2.1		7C	set 2.00	hex 2.0	
75	or a	hex 2.0		7D	set 2.00	hex 2.0	
76	or hl	hex 2.00		7E	set 2.00	hex 2.0	
77	or a	hex 2.0		7F	set 2.00	hex 2.0	
78	or h	hex 2.0		80	set 2.00	hex 2.0	
79	or l	hex 2.1		81	set 2.00	hex 2.0	
7A	or a	hex 2.0		82	set 2.00	hex 2.0	
7B	or h	hex 2.0		83	set 2.00	hex 2.0	
7C	or l	hex 2.1		84	set 2.00	hex 2.0	
7D	or a	hex 2.0		85	set 2.00	hex 2.0	
7E	or hl	hex 2.00		86	set 2.00	hex 2.0	
7F	or a	hex 2.0		87	set 2.00	hex 2.0	
80	or h	hex 2.0		88	set 2.00	hex 2.0	
81	or l	hex 2.1		89	set 2.00	hex 2.0	
82	or a	hex 2.0		8A	set 2.00	hex 2.0	
83	or h	hex 2.0		8B	set 2.00	hex 2.0	
84	or l	hex 2.1		8C	set 2.00	hex 2.0	
85	or a	hex 2.0		8D	set 2.00	hex 2.0	
86	or hl	hex 2.00		8E	set 2.00	hex 2.0	
87	or a	hex 2.0		8F	set 2.00	hex 2.0	
88	or h	hex 2.0		90	set 2.00	hex 2.0	
89	or l	hex 2.1		91	set 2.00	hex 2.0	
8A	or a	hex 2.0		92	set 2.00	hex 2.0	
8B	or h	hex 2.0		93	set 2.00	hex 2.0	
8C	or l	hex 2.1		94	set 2.00	hex 2.0	
8D	or a	hex 2.0		95	set 2.00	hex 2.0	
8E	or hl	hex 2.00		96	set 2.00	hex 2.0	
8F	or a	hex 2.0		97	set 2.00	hex 2.0	
90	or h	hex 2.0		98	set 2.00	hex 2.0	
91	or l	hex 2.1		99	set 2.00	hex 2.0	
92	or a	hex 2.0		9A	set 2.00	hex 2.0	
93	or h	hex 2.0		9B	set 2.00	hex 2.0	
94	or l	hex 2.1		9C	set 2.00	hex 2.0	
95	or a	hex 2.0		9D	set 2.00	hex 2.0	
96	or hl	hex 2.00		9E	set 2.00	hex 2.0	
97	or a	hex 2.0		9F	set 2.00	hex 2.0	
98	or h	hex 2.0		100	set 2.00	hex 2.0	
99	or l	hex 2.1		101	set 2.00	hex 2.0	
9A	or a	hex 2.0		102	set 2.00	hex 2.0	
9B	or h	hex 2.0		103	set 2.00	hex 2.0	
9C	or l	hex 2.1		104	set 2.00	hex 2.0	
9D	or a	hex 2.0		105	set 2.00	hex 2.0	
9E	or hl	hex 2.00		106	set 2.00	hex 2.0	
9F	or a	hex 2.0		107	set 2.00	hex 2.0	
100	or h	hex 2.0		108	set 2.00	hex 2.0	
101	or l	hex 2.1		109	set 2.00	hex 2.0	
102	or a	hex 2.0		10A	set 2.00	hex 2.0	
103	or h	hex 2.0		10B	set 2.00	hex 2.0	
104	or l	hex 2.1		10C	set 2.00	hex 2.0	
105	or a	hex 2.0		10D	set 2.00	hex 2.0	
106	or hl	hex 2.00		10E	set 2.00	hex 2.0	
107	or a	hex 2.0		10F	set 2.00	hex 2.0	
108	or h	hex 2.0		110	set 2.00	hex 2.0	
109	or l	hex 2.1		111	set 2.00	hex 2.0	
10A	or a	hex 2.0		112	set 2.00	hex 2.0	
10B	or h	hex 2.0		113	set 2.00	hex 2.0	
10C	or l	hex 2.1		114	set 2.00	hex 2.0	
10D	or a	hex 2.0		115	set 2.00	hex 2.0	
10E	or hl	hex 2.00		116	set 2.00	hex 2.0	
10F	or a	hex 2.0		117	set 2.00	hex 2.0	
110	or h	hex 2.0		118	set 2.00	hex 2.0	
111	or l	hex 2.1		119	set 2.00	hex 2.0	
112	or a	hex 2.0		11A	set 2.00	hex 2.0	
113	or h	hex 2.0		11B	set 2.00	hex 2.0	
114	or l	hex 2.1		11C	set 2.00	hex 2.0	
115	or a	hex 2.0		11D	set 2.00	hex 2.0	
116	or hl	hex 2.00		11E	set 2.00	hex 2.0	
117	or a	hex 2.0		11F	set 2.00	hex 2.0	
118	or h	hex 2.0		120	set 2.00	hex 2.0	
119	or l	hex 2.1		121	set 2.00	hex 2.0	
11A	or a	hex 2.0		122	set 2.00	hex 2.0	
11B							

35 —	Fixed length string of 32 spaces, used in formatting.
HS —	Hex characters '0'-'F' — also a local variable used in the Hex-Decimal conversion function, FN H (line 50).
LP —	'Flag' set to 1 if printout is required.
D8C —	'Flag' set to 1 if numbers must be output in decimal.
CHECK INDEX —	Line number of the routine which checks to see whether an operation could involve IX or IY.
GET INSTRUCTION —	Line number of the routine which formats a complete line of disassembler output.
MAKE TEXT —	Line number of the routine which formats a complete line of disassembler output.
BYTE VALUE —	Line number of a routine which expresses the contents of C (0-255) in C\$, using the current base.
WORD VALUE —	Line number of a routine which sets up C\$ with a string copy of C (0-65535) in the current base.
RS —	String containing register names.
CS —	String array containing the speeds test.
L,K,T —	Loop counters and temporary values.
AS —	The command typed in by the user.
CS —	The first character of the command.
SUB —	Line number of the chosen monitor subroutine.
LCC —	The location being examined by the monitor.
LS —	The line of text to be output by the monitor.
IO,II,IZ —	The instruction code and its operands.
NS —	The name of the current index register.
SS —	The name of the current indirect register (NI).
MS —	The mnemonic form of the instruction.
NBYTES —	Length of instruction, in bytes.
INDEX —	Set to 1 if IX or IY are to replace HL.
INDIRECT —	Set to 1 if (IX) or (IY) are to replace (HL).
RS —	Character within instruction mnemonic.
MODE —	Addressing mode 0-9; declares number and format of operands.
C —	Number for conversion into a decimal or hex string.
C\$ —	Number after conversion into a string.
DS —	Part of disassembler output line.

Table 1. Variables used in Spectramon.

same first two characters, in some versions of Microsoft BASIC, rather variable names would be allowed since they both contain the key-word **IN**. Sinclair BASIC is also unusual in that it allows spaces to occur in variable names. Table 1 shows all the variable names used in Spectramon and documents their usage.

Other systems can ignore the lines using **COPY** to send out a listing and simply **LPRINT** LS if **LP=1**, printing out lines one at a time rather than *en masse*. A user defined function is set up in line 50, but it is fairly easy to code around this if your computer doesn't support that feature. **FNH(H)** simply returns the decimal value of the first character in **HS** — 1 for 'T', 10 for 'A', 11 for 'B' and so on.

Spectramon uses a few **PEEKs** and **POKEs** which will not be required on other systems. **POKE 23688** is a useful command which forces the Spectrum into capitals-lock (selecting a flashing 'C' as a cursor rather than a flashing 'L'). This ensures that commands are entered in capitals (unless the user purposely switches to lower-case in the course of entering a command). The location 23689 contains the number of empty lines on the Spectrum screen — when **POKE 23689** is three or less the screen is assumed to be full since the bottom two lines aren't normally used for text and a line is needed for the 'More?'... message.

Location 13260 contains the ASCII code of the key most recently pressed. It is set to 32 when the space bar has been pressed (or is being simulated) and 13 when Enter has been typed.

THE LAST WORD...

When I received my Spectrum I was convinced that I'd never get used to the keyboard. After writing, editing and typing in Spectramon I was well-practiced! Hopefully, the program also illustrates a few useful tricks of ZX BASIC, both from the BASIC and the assembler programmer's point of view.


```

1600 B BIT INDIRECT: BRANCH INST.
1610 GO SUB GET_INDIRECT=2
1620 LET INDEX=0; LET INDIRECT=0
1630 REM PREPARE IX & IY AGAIN
1640
1650 IF HLEN=7 THEN GO SUB CHECK
1660 REM HLEN=8/9
1670 LET MYTES=MYTES+3; INDIRECT
1680
1690 GO SUB MAKE TEXT
1700 GO TO 1110
1710 LET INDIRECT=0; LET INDEX=0
1720 LET I=0
1730 IF I=I+1; IF I=LEN H8 THEN
RETURN
1740 LET HLEN=I+1; IF HLEN=1; IF HLEN=2
AND HLEN=3 THEN GO TO 1810;
REM CHECK FOR address aliases
1750 IF HLEN=1 THEN GO TO 1800
1760 LET INDEX=1; REM use IY only
1770 LET HLEN=2; REM HLEN=3
1780 LET I=HLEN; REM HLEN=1
1790 LET I=I+1; REM HLEN=2
1800 RETURN
1810 LET INDIRECT=1; REM use IX
1820 LET I=I+1; IF HLEN=1; IF HLEN=2
AND HLEN=3 THEN GO TO 1810;
REM CHECK FOR address aliases
1830 IF HLEN=1 THEN GO TO 1800
1840 LET INDEX=1; REM use IY only
1850 LET I=HLEN; REM HLEN=1
1860 LET I=I+1; REM HLEN=2
1870 RETURN
1880 LET I=HLEN+1; REM Get to
1890 IF OPCODE=1 THEN LET
HLEN=HLEN+1; GO TO 2000
1900 REM Check for trailing spaces
1910 LET HLEN=HLEN-1; REM use
addressing mode 1/2 SPECIAL
1920 IF HLEN=1 OR HLEN=2 THEN LET
HLEN=HLEN+1; REM not special
1930
1940 LET I=HLEN; LET I=I+1; REM
skip off the end marker
1950 FOR I=1 TO LEN H8; IF HLEN=1
TO I; THEN GO TO 2000; REM F
in operands field
1960 NEXT I
1970 LET H8=H8-I+1; GO TO 2-LEN H8
1980 REM no operands, repeat really
1990 LET H8=0; REM Operands!
2000 GO TO 2005
2010 LET H8=H8+1; I=I+1; GO TO 2
2020 REM H8=1 is the opcode
2030 LET H8=H8+1; GO TO LEN H8; REM
and I=1 is the operand
2040 GO TO 2000+HLEN*1000; REM Fy
operand in accordance with addr op
2050
2060 REM 11 Convert hex to dec
2070 IF LEN H8 THEN GO TO 2080
2080 REM HLEN=1; LET H8=0
2090 LET H8=H8*16+INT H8/16; REM
2100 LET H8=H8*16+H8%16; GO TO LEN H8
2110 LET H8=H8*16+H8%16; GO TO LEN H8
2120 REM force a character read
2130 LET LOC=LOC; FOR I=0 TO 3; LE
T H8=H8*16+INT H8/16; REM 1; GO TO 2080; REM Get va
lue
2140 REM use addressing mode
2150 LET MYTES=1; REM loaded a
character - no operand bytes
2160 REM HLEN=0; REM Built out
the instruction text
2170 RETURN
2180 LET MYTES=1; REM 8 bit ope
rand in or pointed to by register
2190 IF LEN H8=1 THEN LET H8=
H8*16+INT H8/16; REM 1
2200 LET H8=H8*16+H8%16; REM
get operand number from inst
2210 LET H8=H8+H8*16; REM add
operands bytes
2220 IF H8=1 THEN LET H8=H8*16
2230 LET H8=H8*16+H8%16; REM
2240 RETURN
2250 LET MYTES=2; REM 8 bit ope
rand if operand
2260 IF LEN H8=1 THEN LET H8=
H8*16+INT H8/16; REM 1
2270 GO SUB GET_NUMBER
2280 GO SUB GET_VALUE

```

```

0040 LET H$=H$+B$+C$; REM BUILD
0050 RETURN
0060 LET BYTES=3; REM 16 bit
0070 IF C=0FFFF THEN
0080 IF LEN (B$) < 3 THEN LET B$=
0090 LET C=255+12+11
0100 GO SUB C=00 VALUE
0110 LET H$=B$+B$+C$
0120 RETURN
0130 LET BYTES=2; REM Relative
0140 IF C=1 IS OFFEND
0150 IF LEN (B$) < 2 THEN LET B$=
0160 LET C=LOC-254+11; REM MAX
0170 ABSOLUTE 255; FROM OFFSET
0180 IF 11:255 THEN LET C=C+255
0190 REM C=0 255 or 15554
0200 GO TO 2000; REM Display and
0210 END; END; END; END; END; END;
0220 LET BYTES=3; REM 32 bit
0230 LET C=11
0240 GO SUB BYTE VALUE
0250 GO TO 2000
0260 LET BYTES=3; REM Indirect
0270 IF 11:255 THEN
0280 GO SUB WORD VALUE
0290 IF LEN (B$) < 3 THEN LET B$=
0300 LET H$=B$+B$+B$+C$+1; REM
0310 B$+C$+1 indicates indirect
0320 LET BYTES=2; REM IN 16 bit
0330 IF 11:255 THEN
0340 LET C=11
0350 GO SUB BYTE VALUE
0360 IF 11:255 THEN
0370 LET BYTES=3; REM STORE at
0380 OFFEND address
0390 LET C=255+12+11
0400 GO SUB WORD VALUE
0410 IF LEN (B$) < 2 THEN LET B$=
0420 LET H$=B$+B$+C$+1; REM
0430 B$+C$+1 indicates address not
0440 valid
0450 RETURN
0460 LET BYTES=1; REM Peculiar
0470 LET H$="?"
0480 RETURN
0490 REM 255 Build up 2/3 line
0500 LET C=LOC; GO SUB WORD ONLY
0510 IF 000 THEN LET B=C$+" ";
0520 GO TO 1000
0530 LET B=C$+11:255 TO 1111; REM
0540 B$+11:255 TO 1111 line
0550 LET C=1; FROM T-LOC TO LOC
0560 BYTES=1
0570 LET C=DEC+7; REM Set data
0580 IF 000 THEN LET C=C+8; GO S
0590 BYTES VALUE; LET C=C+1; GO TO
0600 REM Always use hex number
0610 GO SUB BYTE VALUE
0620 LET C=C+B+C+1 TO 20; REM H
0630 B$=B$+B$+C$
0640 REM 255 REM Process all the
0650 B$+11:255 TO 1111 line
0660 LET B=C$+C+1 TO 254+11
0670 BYTES=1; REM End space 1111
0680 LET LOC=LOC+BYTES; REM Adv
0690 255 255 instruction
0700 RETURN
0710 REM 255 Convert C to C$
0720 IF 000 THEN LET C$=C+1;
0730 REM Decal 11:255
0740 LET C$=C+11:255 TO 215+1;
0750 C$=C+11:255 TO 215+1; C$=
0760 C$+11:255 TO 215+1; RETURN;
0770 IF 000 THEN
0780 IF 000 THEN LET C$=C$+1;
0790 LET C$=C$+1;
0800 LET C$=C$+1;
0810 LET C$=C$+1;
0820 LET C$=C$+1;
0830 LET C$=C$+1;
0840 LET C$=C$+1;
0850 LET C$=C$+1;
0860 LET C$=C$+1;
0870 LET C$=C$+1;
0880 LET C$=C$+1;
0890 LET C$=C$+1;
0900 LET C$=C$+1;
0910 LET C$=C$+1;
0920 LET C$=C$+1;
0930 LET C$=C$+1;
0940 LET C$=C$+1;
0950 LET C$=C$+1;
0960 LET C$=C$+1;
0970 LET C$=C$+1;
0980 LET C$=C$+1;
0990 LET C$=C$+1;
1000 LET C$=C$+1;
1010 LET C$=C$+1;
1020 LET C$=C$+1;
1030 LET C$=C$+1;
1040 LET C$=C$+1;
1050 LET C$=C$+1;
1060 LET C$=C$+1;
1070 LET C$=C$+1;
1080 LET C$=C$+1;
1090 LET C$=C$+1;
1100 LET C$=C$+1;
1110 LET C$=C$+1;
1120 LET C$=C$+1;
1130 LET C$=C$+1;
1140 LET C$=C$+1;
1150 LET C$=C$+1;
1160 LET C$=C$+1;
1170 LET C$=C$+1;
1180 LET C$=C$+1;
1190 LET C$=C$+1;
1200 LET C$=C$+1;

```

```

0000 LET CT=C-INT IC/250000000.0
01 CT=C-INT CT/100000000.0 TO CT/100000000.0
02 LET C=C-INT CT/100000000.0
03 REM 50 mil has value
04000 REM == 100 instructions
05000 DATA "CODELN"
06000 DATA "PROP"
07000 DATA "LD BC"
08000 DATA "LD B"
09000 DATA "LD C"
10000 DATA "LD A"
11000 DATA "LD X"
12000 DATA "LD H"
13000 DATA "LD D"
14000 DATA "LD E"
15000 DATA "LD I"
16000 DATA "LD J"
17000 DATA "LD K"
18000 DATA "LD L"
19000 DATA "LD M"
20000 DATA "LD N"
21000 DATA "LD O"
22000 DATA "LD P"
23000 DATA "LD Q"
24000 DATA "LD R"
25000 DATA "LD S"
26000 DATA "LD T"
27000 DATA "LD U"
28000 DATA "LD V"
29000 DATA "LD W"
30000 DATA "LD X"
31000 DATA "LD Y"
32000 DATA "LD Z"
33000 DATA "LD A"
34000 DATA "LD B"
35000 DATA "LD C"
36000 DATA "LD D"
37000 DATA "LD E"
38000 DATA "LD F"
39000 DATA "LD G"
40000 DATA "LD H"
41000 DATA "LD I"
42000 DATA "LD J"
43000 DATA "LD K"
44000 DATA "LD L"
45000 DATA "LD M"
46000 DATA "LD N"
47000 DATA "LD O"
48000 DATA "LD P"
49000 DATA "LD Q"
50000 DATA "LD R"
51000 DATA "LD S"
52000 DATA "LD T"
53000 DATA "LD U"
54000 DATA "LD V"
55000 DATA "LD W"
56000 DATA "LD X"
57000 DATA "LD Y"
58000 DATA "LD Z"
59000 DATA "LD A"
60000 DATA "LD B"
61000 DATA "LD C"
62000 DATA "LD D"
63000 DATA "LD E"
64000 DATA "LD F"
65000 DATA "LD G"
66000 DATA "LD H"
67000 DATA "LD I"
68000 DATA "LD J"
69000 DATA "LD K"
70000 DATA "LD L"
71000 DATA "LD M"
72000 DATA "LD N"
73000 DATA "LD O"
74000 DATA "LD P"
75000 DATA "LD Q"
76000 DATA "LD R"
77000 DATA "LD S"
78000 DATA "LD T"
79000 DATA "LD U"
80000 DATA "LD V"
81000 DATA "LD W"
82000 DATA "LD X"
83000 DATA "LD Y"
84000 DATA "LD Z"
85000 DATA "LD A"
86000 DATA "LD B"
87000 DATA "LD C"
88000 DATA "LD D"
89000 DATA "LD E"
90000 DATA "LD F"
91000 DATA "LD G"
92000 DATA "LD H"
93000 DATA "LD I"
94000 DATA "LD J"
95000 DATA "LD K"
96000 DATA "LD L"
97000 DATA "LD M"
98000 DATA "LD N"
99000 DATA "LD O"

```

[illegible]

SKULL

It's a monster...

...and it's waiting for you

ENTER THE CATACOMBS...
HUNT THE TREASURE...
AVOID THE TRAPS...
WATCH OUT FOR SKULL!

SUPER NEW, ORIGINAL, 3-DIMENSIONAL
FULL-GRAPHICS MAZE GAME FOR THE SPECTRUM 48K.
HUNT THE TREASURE THROUGH MULTI-LEVEL MAZES
AVOIDING HAZARDS AND DEADLY WANDERING SKULLS.
*THIS GAME WILL BLOW YOUR MIND.
THERE IS NOTHING ELSE LIKE IT!*

Write for details on the GRAND HIGH SCORE COMPETITION
GAMES MACHINE LTD.,
40 PRETHERNE ROAD, WELWYN GARDEN CITY, HERTFORDSHIRE.
☎ (07673) 28444/28426

GAMES MACHINE



WHERE THE FUN BEGINS...

IT WILL DRIVE YOU... **BONKERS**



PROCOM SOFTWARE

FIRST ACTION
ARCADE GAME

FOR ANY ZX SPECTRUM

BONKERS

Try to escape from the outer regions of an alien space station... Fight the deadly Pac-Manoids and Baby Glands... but do not relax... there are other alien weapons!
• Arcade Action!
• High Score prize!
for any ZX Spectrum

£5.50

CODE OF THE YEAR 1987
WINNER

GRID MASTER



PROCOM SOFTWARE

WORKS ON 48K ZX SPECTRUM

GRID MASTER

A game for all the family to enjoy. It does not even get tedious... or does it? Lots of fun!

£4.95

TIME IS RUNNING OUT!



BREAK AWAY

by PROCOM
Runs on ZX Spectrum hardware

BREAK AWAY

Be the modern Robin Hood! Time hasn't, you have 15 seconds to rescue the dammed in distress... Can you do it? Try! Fun for all the family for ZX Spectrum 128K and

£5.50

FOR THE ZX SPECTRUM

PROCOM SOFTWARE

* Bonkers and Breakaway are available from Primus, Woolworths, Rumbelows, John Lewis and Debenhams.

* Dealer inquiries welcomed

* Overseas inquiries welcomed

DO YOU HAVE A GOOD IDEA FOR A GAME? CALL US!

COMING SOON!

- * Bricky Micky
- * Bermuda Triangle
- * Time Bug
- * Magic Orchard
- * Final Frontier

ask for details -

HOW TO ORDER

Telephone orders
(011) 508 1218

Quoted prices include Postage and Packing and are subject to change without notice. Please note that prices are subject to change without notice. Please note that prices are subject to change without notice.

BY POST

Simply send your requirements on the order form, and we will charge you by value of your order. Please note that prices are subject to change without notice. Please note that prices are subject to change without notice.

IN PERSON

Visit our premises, at the address below for a great selection - we will be delighted to see you.

POST TO: PROCOM SOFTWARE, 309 HIGH ROAD, LOUGHBOROUGH, ESSX.

Please send me the following items:

- ☐ Bonkers £5.50
- ☐ Grid Master £4.95
- ☐ Breakaway £5.50
- ☐ List of new games

I enclose my cheque for £...
I prefer to pay with my ACCESS/BACARD card.
(Credit cards must be paid by cash on delivery)

Company
Signature
Name
Address

Tel: 05341 Tel: 05341

Could cards with it please be sent to the address above. Please note that prices are subject to change without notice.

PROCOM SOFTWARE

PROCOM SOFTWARE
309 HIGH ROAD - LOUGHBOROUGH - ESSX
TEL: (011) 508 1218