An Argus Specialist Publication
WINTER 1984
£2.95

THE ZX Collection

- PAGES PACKED WITH SUPER PROGRAMS FOR THE SPECTRUM AND ZX81
- GREAT UTILITIES AND USEFUL ROUTINES TO MAKE LIFE EASIER
- SPACE GAMES, ADVENTURES, BOARD AND STRATEGY GAMES FOR YOU TO PLAY

- READERS GIVE THEIR OWN VIEWS OF THE SOFTWARE AROUND

MIND GAMES

AMERICAN FOOTBALL

A fast & furious graphics strategic simulation

£9.99
FREE RULE BOOK

FOR THE SPECTRUM 48K CBM 64

OTHER GREAT MIND GAMES:

Starring The Overlords of the Universe
The candidate (you) have to get to the Chamber of Creation. It's a laugh a minute, since it's 2,000 light years away on the most horrible planet in the Universe ... and your starship doesn't work either!

A full feature adventure starring well known nasty aliens the Zarps. Can you play the hero and stop their plans to blow up the earth.

Starring The Zurgs
After a desperate space battle only one fleet of heroes remain to prevent the invasion of earth. The future of humanity lies with you!

Argus Press Software Group

For mail order, write with cheque/P.O./card No. to:
Mind Games, Argus Press Software Group, No.1 Golden Square, London W1A 3AB

BY WENDY J PALMER

FOREWORD

It is at this time of the year when frustrated parents search the local stores for those last-minute Christmas gifts for loved ones and friends. Well THIS year their problems are over — here is the perfect gift both for the young and not-quite-so-young who have access to either a ZX81 or Spectrum computer — THE ZX COLLECTION.

Not Only at Christmas

Throughout the year Argus Specialist Publications publish the highly popular magazine for the Sinclair micro owner, ZX Computing. But this year we thought we'd give panicking parents and Santa Claus a helping hand and by compiling a package of original programs, articles and reviews for the ZX81 and Spectrum user. Technical expertise has been provided by the Editor of ZX Computing who has also tested all of the programs to be found in these pages. If you follow the instructions given in the articles, you should have no problem getting the programs to work. However, should you find you do come up against any difficulties, please write to us at the editorial address given on the Contents page, stating very clearly exactly what your problem is. It is almost impossible for us to answer technical enquiries on the telephone, so please write and don't ring in. Also write if you would like any information on ZX Computing itself.

Feeling Content?

We have tried to cater to as many different tastes in THE ZX COLLECTION as possible and we hope that there is something to interest most of you, whether you have just unpacked your computer from its Christmas wrapping or whether you are an 'old hand' looking for new stimulation.

The games are exciting and taxing — you can either save the world and destroy the aliens in a number of the programs such as Andromeda 3, Space Rescue and Aliens or you may prefer the more quieter and less violent pursuits of games such as Dominoes or Codebreaker. You can learn to land an aircraft safely, manoeuvre the suicidal frog along a danger-ridden route or unload cargo from a docked ship before the helicopter steals it from you. There are these and more games to test your wits and reflexes in THE ZX COLLECTION.

Not all computer owners and users are games players (well not all of the time anyway) but you need not despair — you will find a number of useful routines, programs and ideas within these pages. We look at writing data base programs, how the Spectrum can analyse sound, how you can create extra graphics and there's even an O/S Disassembler. You can learn about astronomical constellations, maths and French.

Great Value

There's no doubt that this gathering of programs together with articles and reviews written by ZX Computing readers to guide you through the vast amount of commercially available software around, constitutes a great buy for ZX81 and Spectrum owners, games enthusiasts and serious users alike. We've enjoyed putting the magazine together and we hope that you will gain many hours of information, help and plain, old-fashioned fun now that youve bought it.

Oh and by the way, Merry Christmas and a Happy New Year to all of our readers!

ZX Collection

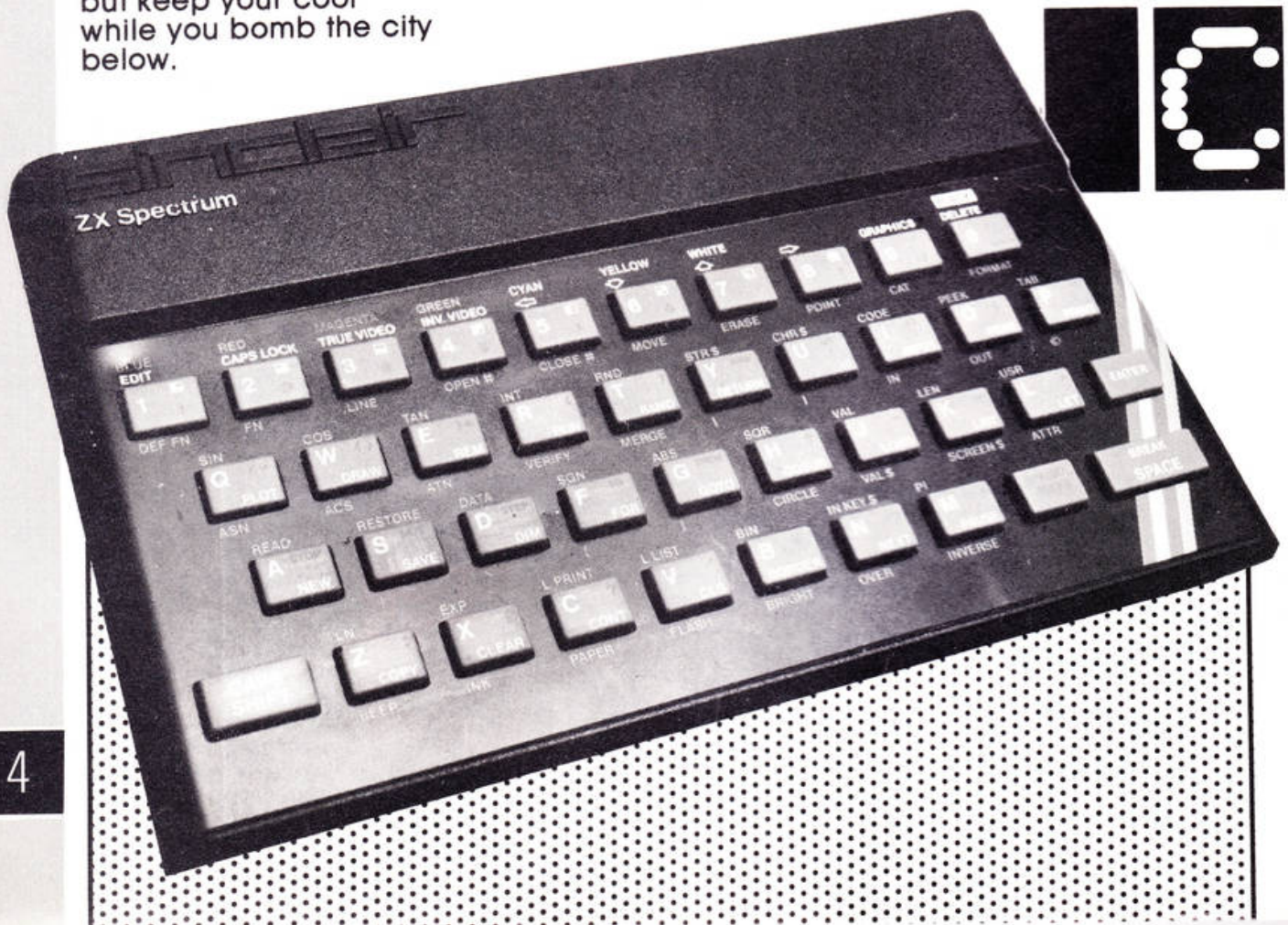
EDITOR: WENDY J PALMER
CONSULTANT EDITOR: RAY ELDER
PRODUCTION ASSISTANT: JANICE HICKS
ADVERTISEMENT MANAGER: BARRY BINGHAM
CHAIRMAN: JIM CONNELL
ORIGINATION AND DESIGN: MM DESIGN, LONDON

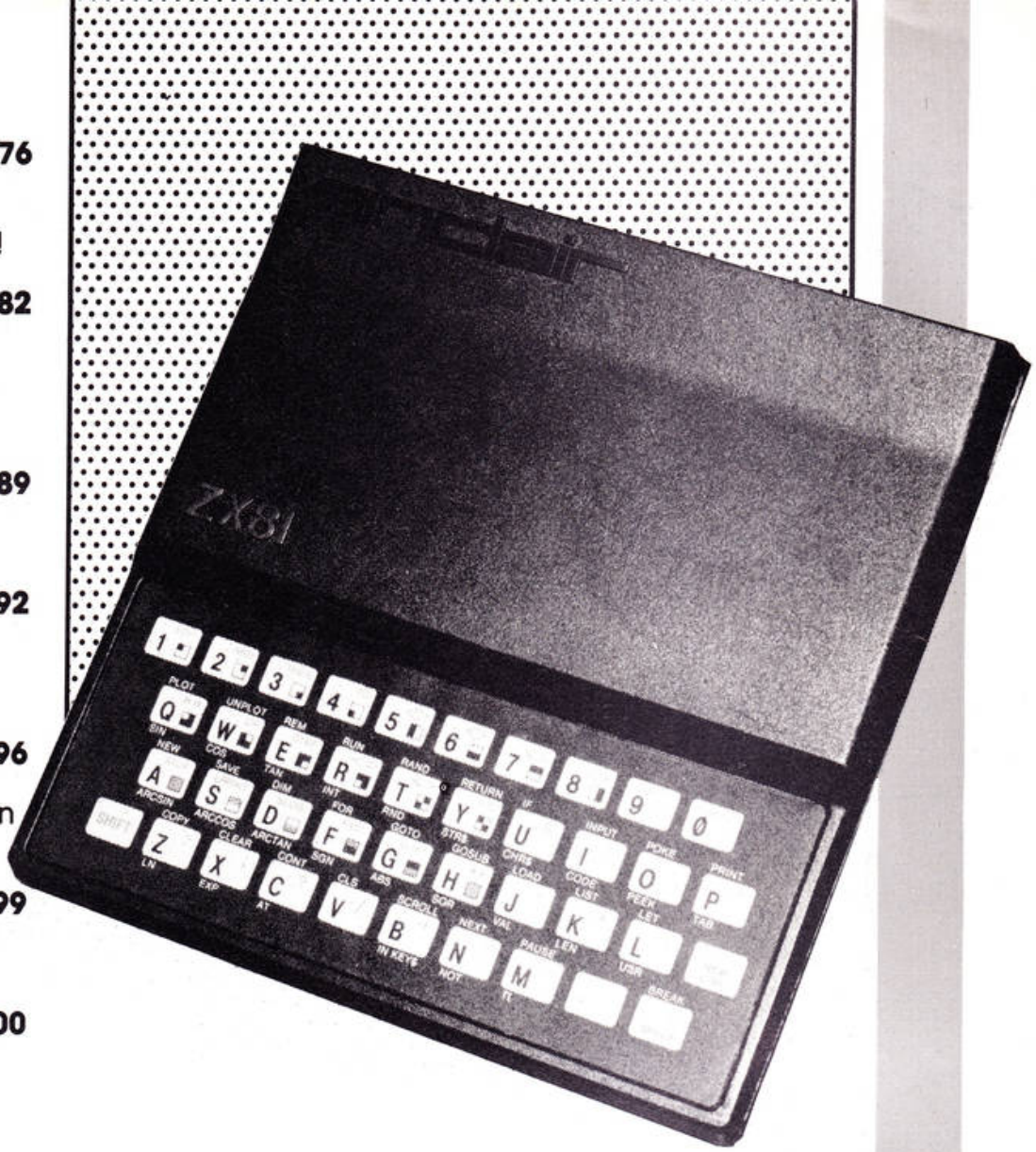
Distribution by Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS.

Printed by Ambassador Press

EDITORIAL & ADVERTISEMENT OFFICE, No. 1 GOLDEN SQUARE, LONDON. W1R 3AB.
TELEPHONE: 01-437 0626
TELEX: 8811896.

WINTER 1984





CONTENTS

If you think that you're the adventurous type, just have a go at escaping from Andromeda 3!

ANDROMEDA 3

BY GRAEME POOLE

The time is 12.15 am in the year 3058 AD. You were on a routine flight to all of the major planets in the Solar System. Suddenly, out of nowhere, appeared five Space Fighters from Andromeda 3. Unfortunately your instruments did not register them and you were taken completely by surprise.

The fighters launched a full-scale attack on your ship (for no apparent reason) and your shipped was crippled beyond normal flight. You did, however, have enough of the basic controls left working so that you could crash land your spaceship on the vast planet of Andromeda 3. Your ship was completely destroyed but you managed to escape, alive and without injury.

By this time, the Andromedan fighters had landed and the pilots took you to the leader of the planet. Before you had a chance to speak, the pilots explained to the leader that you had launched the attack on THEM. You of course denied this, but unfortunately, the leader did not believe you. He ordered you to be taken into the depths of the planet, where prisoners are left to rot in a large underground prison.

It is here in this prison cell that you are now sitting; you must try to escape alive and report back to Earth notifying them of the situation. This is of the utmost importance: you have heard two guards talking in the guardroom, saying that the leader was going to report the 'incident' to Earth and that full-scale war was to be launched between the two planets.

So you can see that the entire future of Earth depends on you... Good Luck!

Playing the Game

You move around the prison by choosing the direction you wish to go. There are 10 possible directions you can try. They are as follows: N, NE, E, SE, S, SW, W, NW, U and D. As you move around, you will be given a description of your location along with a picture of what you can see. On your travels you will find various objects in certain rooms and you can either TAKE or GET these.

There are 28 different instructions you can use to talk to the computer. They are: N, NE, E, SE, S, SW, W, NW, U, D, LIGHT, UNLOCK, USE, OPEN, CROSS, JUMP, GET, TAKE, BRIBE, GIVE, LIST, SWIM, FEED, FIRE, PUT, SHOOT, SHOW and EAT. There are 2 commands which will probably be very useful to you, these are LIST and Z. If you type LIST, you will be given a complete list of the objects you are carrying if you have any. Typing Z (the key with COPY on it) the computer will copy the screen to the printer which you can then use to create a map as you go along. This saves a lot of time if you want a map with the pictures and detailed descriptions.

The program, unlike other adventures, does not restrict you to using only two words like GET TORCH etc, but will allow a fair amount of different variations of one sentence to make it more 'human' and friendly. For example, all the following sentences will result in the same command being executed: JUMP PIT, JUMP THE PIT, JUMP OVER PIT, JUMP OVER THE PIT, JUMP ACROSS THE PIT etc. There is only one location in which you can be killed, and you are given plenty of warning about this, so that you will not have to restart every 30 seconds or so. There are over 70 locations to explore, but you may not have to explore them all to escape from the prison, but then again you might. There are lots of one way passages making the game more difficult, and also beware of the twisting paths which can take you from one side of the prison to the other in one move which can be very annoying.

The Program

The user defined graphics and all the variables used in the program are created in lines 20 to 50. Lines 100 to 3650 contain all the information for the 72 locations along with some special variables to tell the computer where you are and what to do depending on what you type in

The main part of the program is from lines 9500 to 9930 This checks the input and moves you from room to room or carries out specific instructions like OPEN DOOR It also checks that you have taken an object before you try to use it, and if you try to take an object it checks to see if the object is in the room. Most of the computers replies are held in lines 3700 to 4500 and each one is printed depending on where you are and what you asked the computer to do.

Each of the room descriptions has a string variable called r$. This string is used to hold the line numbers of the rooms in the 10 possible directions, and they are held in the order N,NE,E,SE,S,SW,W,NW,U, D. For example, if r$= "32501000 1250", then going North would take you to line 3250, NE would go to 1000 and East would be 1250. If r$ = "170020500000", then North is 1700, NE is 2050 and East is a wall. Walls are detected by 0000. Line numbers which are not 4 digits long, eg 800 are written as 0800 etc.

Lines 9505, 9660, 9670 and 9680 convert the input into numbers and pick out the relative set of numbers from r$. The user can convert this program to make his own adventures quite easily by changing the room descriptions, the relevant r$, and the objects used. The adventure was originally written as an entirely text only game. But I felt that the addition of some simple graphics colour and sound added more interest to the game.

```
1 REM ************************
      #Underlined characters#
      #are entered in       #
      #GRAPHICS mode.       #
      ************************
 16 CLS : BORDER 7: PAPER 7: CL
S : INK 0: PRINT "CREATING UDG'S
  PLEASE WAIT"
 20 FOR q=1 TO 19: READ a$: FOR
g=0 TO 7: READ a: POKE USR (a$)
+g,a: NEXT g: NEXT q: GO TO 30
 21 DATA "a",0,0,255,128,128,2
5 0,0,"b",1,231,249,1,1,249,1,1,
"c",3,12,16,16,127,32,16,15,"d",
192,48,8,8,254,4,8,240
 22 DATA "e",24,60,126,231,231,
126,60,24,"f",0,0,0,255,255,14,
0,10,"m",3,14,56,33,33,56,14,3,"
n",192,112,28,132,132,28,112,192
 23 DATA "s",255,255,102,102,10
2,102,255,255,"o",1,7,12,24,12,
,1,0,"p",192,112,24,12,24,112,19
2,0,"q",2,229,63,112,127,116,124
,112,"r",131,66,255,1,255,0,0,0
 24 DATA "g",0,BIN 01010101,BIN
 00101010,64,32,BIN 01001110,BIN
 00101010,BIN 01001110,"h",0,BIN
 01010101,BIN 10101010,0,0,BIN 1
1101110,BIN 10101000,BIN 11101
0
 25 DATA "i",0,BIN 01010100,BIN
 10101010,4,2,BIN 11100100,BIN
0000010,BIN 11100100,"j",BIN 00
01000,BIN 01001000,32,64,32,BIN
 01010101,BIN 00101010,0
 26 DATA "k",BIN 10100010,BIN
 0101110,0,0,0,BIN 01010101,BIN
 0101010,0,"l",BIN 00100010,BIN
 11100100,2,4,2,BIN 01010100,BIN
 0101010,0
 30 FOR G=1 TO 400: NEXT G: PRI
NT : PRINT "Please put the caps
lock on": INPUT i$
 40 LET padr=6100: LET passage=
6000: LET rp=5e3: LET rp2=5100:
LET door=5200: LET doorr=5300:
ET door1=5400: LET hole=5500: LE
T cave=5600: LET cdr1=5700: LET
cdrr=5800: LET cdrf=5900
 50 DIM a$(32): LET key=0: LET
torch=key: LET torch=key: LET d
agger=key: LET money=key: LET food
=key: LET pos=key: LET ladder=ke
y: LET pass=key: LET laser=key:
LET NX=9500: POKE 23609,30: LET
RM=600: GO TO 9690
 100 PRINT "You are in a small r
oom with a  door to the east. Th
e only  way  out is to the south.
": LET r$="0000000000000000002500
0000000000000000000"
 101 PRINT AT 19,0;"The door is
 "; "OPENED" AND key(32)+("OPEN
" AND key<>2)
 102 LET pos=1: GO SUB rp: GO SU
B doorr: GO TO NX
 150 PRINT "You are now in a sma
ll square  room. You can see 3
paths which go east, west and do
wn ": LET r$="0000000002000000000
0000010000000000950"
 151 LET pos=0: GO SUB rp: GO SU
B doorr: GO SUB door1: GO SUB ho
le: GO TO NX
 200 PRINT "There is a very big
 ast guard in this room. He doe
sn't look  real pleased to see y
ou  He is  guarding a small dis
k. HINT..  He is Corrupt.": LET
r$="0000000000000000000000000015
0000000000"
 201 LET pos=3: GO SUB rp: GO SU
B door1: GO TO NX
 202 GO SUB rp
 249 GO TO NX
 250 PRINT "You are in a complet
ely dark  room. It is so dark
that you can only see to go the w
ay you came (SE). You need some
light to go any further.": LET r
$="0000000000000035000000000000
000000000"
 251 IF torch=2 THEN  PRINT #1;"
Your torch lights the way.": BEE
P .3,10  PAUSE 75
 252 IF torch=2 THEN  GO SUB rp:
 GO SUB door
 257 LET pos=4: GO SUB rp2: GO T
O NX
 300 PRINT "You are at a flight
of stairs  which lead down. It
doesn't look too  nice down there.
 You can  feel a cold wind blo
wing up the stairs.": LET r$="00
0000000000000005000000000000000
000350"
 301 GO SUB rp2: PLOT 72,104: DR
AW 110,0: DRAW 0,62: DRAW -110,0
: DRAW 0, -62: PLOT 0,175: DRAW 7
2, 0: PLOT 255,175: DRAW -72,-8
 302 PLOT 0,55: DRAW 72,48: PLOT
 255,55: DRAW -72,48
 303 PLOT 64,64: DRAW 126,0: DRA
W -32,30: DRAW -62,0: DRAW -32,-
30: PLOT 88,64: DRAW 28,30: DRAW
 0,-8: DRAW -17,-22
 304 PLOT 116,85: DRAW 13,0: DRA
W -12, -20: PLOT 129,85: DRAW 0,-
8: DRAW -5,-12: PLOT 129,76: DRA
W 13,0: DRAW -3,-11: PLOT 142,76
: DRAW 0,-11
 349 GO TO NX
 350 PRINT "You are now at the b
ottom of a  flight of stairs. Yo
u are in a  huge dungeon. There
are 2 paths,one goes NW and the
other SW.": LET r$="000000000000
0000000045000000250030000000"
 352 LET pos=0: GO SUB rp2: PLOT
 135,56: FOR q=1 TO 7: DRAW 0,16
: DRAW 16,0: NEXT q: PLOT 111,64
: FOR q=1 TO 6: DRAW 0,16: DRAW
16,0: NEXT q: DRAW 0,14: DRAW 16
,0
 353 LET a=111: LET b=64: LET k=
7: GO SUB 355: LET a=111: LET b=
80: LET k=6: GO SUB 355: GO TO 3
58
 355 FOR q=1 TO k: PLOT a,b: LET
 c=a: LET d=b: DRAW 24,-8: LET a
=c: LET b=d: LET a=a+16: LET b=b
+16: NEXT q: RETURN
 358 PLOT 208,174: DRAW 24,-7: P
LOT 224,174: DRAW 23,-6: DRAW 0,
6: PLOT 0,64: DRAW 110,0: GO TO
NX
 400 PRINT "You have walked into
 a dead end.The only way out is
south.": LET r$="000000000000000
0050000000000000000000000"
 448 GO SUB rp: GO TO NX
 450 PRINT "You are in a small c
ave. It is  fairly dark but you
can just seeto go on if you are
```

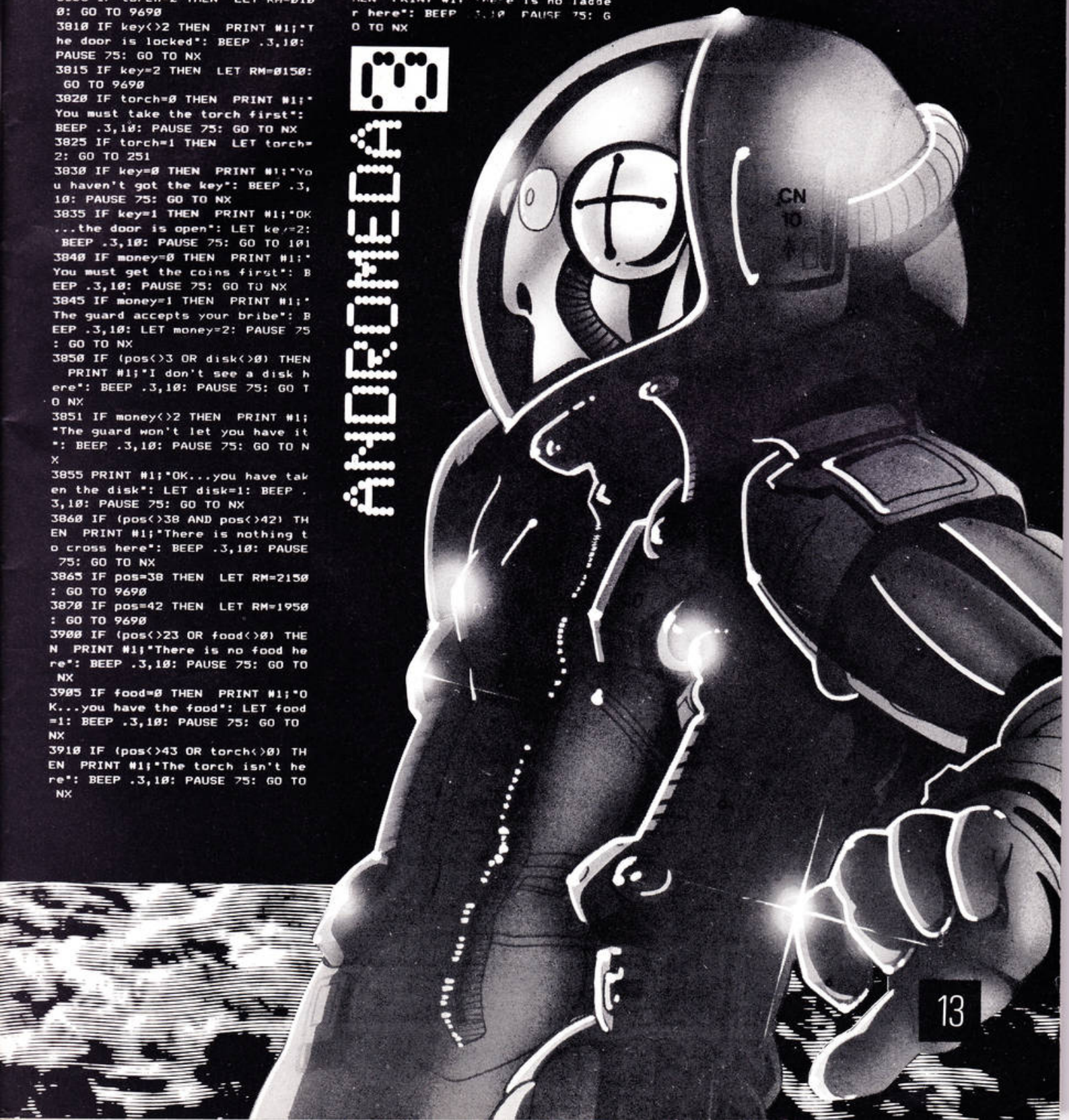
```
careful.": LET r$="000003500000
0000900000000000000000000000"
 452 GO SUB cave: GO TO NX
 500 PRINT "You are now in a sma
ll empty lowchamber. In the dist
ance you cansee a flight of stai
rs leading  down. Paths go north
 and sduth.": LET r$="030000000
0000000950000000000000000000"
 548 GO SUB rp: GO SUB door: GO
TO NX
 550 PRINT "This is the big pris
on corridor.It is totally empty.
 Five paths lead away from here.
 They go SE,SW, south, north and
 east. You  can here snoring fro
m the east.": LET r$="0600000007
501000120009500000000000000000"
 553 LET pos=0: GO SUB passage:
GO SUB padr: PLOT 0,60: DRAW 0,9
0: DRAW 32,-0: DRAW 0,-70: PLOT
247,60: DRAW 0,90: DRAW -32,-8:
DRAW 0,-70
 554 PLOT 64,82: DRAW 0,54: DRAW
 16,-5: DRAW 0,-43: PLOT 191,82:
 DRAW 0,54: DRAW -16,-5: DRAW 0,
-43: GO TO NX
 600 PRINT "You are sitting all
alone in a  small empty prison c
ell. You cansee that your door i
s unlocked. The only passage goe
s to the SW.": LET r$="000000000
0000000000005500000000000000000"
 602 GO SUB rp: PLOT 100,80: DRA
W 0,48: DRAW 32,0: DRAW 0,-48: D
RAW -32,0: PLOT 100,80: DRAW 24,
8: DRAW 0,40: PLOT 16,104: DRAW
0,38: DRAW 38,-8: DRAW 0,-24: DR
AW -38,-7
 603 PLOT 26,106: DRAW 0,34: DRA
W 1,0: DRAW 0,-34: DRAW 1,0: DRA
W 0,34: PLOT 42,109: DRAW 0,27:
DRAW 1,-1: DRAW 0,-26: DRAW 1,0:
 DRAW 0,26: PLOT 114,110: DRAW 1
0,2: DRAW 0,10: DRAW -10,-2: DRA
W 0,-10: PLOT 117,112: DRAW 0,8:
 PLOT 121,111: DRAW 0,9: GO TO N
X
 650 PRINT "You are in a small r
oom used by the prison guards fo
r torturing all the prisoners. Y
ou should gobefore the guards ca
tch you.": LET r$="000000000000
0000000000005500000000000000"
 698 GO SUB rp: GO SUB door1: GO
 TO NX
 700 PRINT "This is a small room
 used by theguards for solitary
confinement of vicious or awkwar
d prisoners.": LET r$="000000000
0000000000000000000000000000000"
 741 GO TO 601
 750 PRINT "You have walked into
 the guard  room. Be very quiet,
 the guards are having 40 winks!
 don't wake them up.": LET r$="0
0000000000000010000000055000000
0000000"
 751 LET pos=14: IF key=0 THEN
PRINT AT 3,12;"You can";AT 5,14;
"see";AT 7,13;"a KEY";AT 13,15;"
EE"
 798 GO SUB rp: GO SUB doorr: GO
 SUB door1: GO TO NX
 800 PRINT "You are now in the i
nterrogationroom. Three passages
 lead out ofhere. They go to the
 north, NE  and west. You can he
re snoring  from the west.": LET
 r$="0650070000000000000000000075
0000000000000"
 848 LET pos=0: GO SUB rp: GO SU
B door: GO SUB door1: GO TO NX
 850 PRINT "This is a small narr
ow passage. You can see a small
room below  through a hole in th
e floor.": LET r$="0000000009000
0000000000000000040000001100"
 898 GO SUB passage: GO SUB hole
: GO TO NX
 900 PRINT "Ybu are in a deserte
d room. you can see a small cave
 not too faraway to the north. 2
 other pathsgo west and SE.": LE
T r$="0450000000001150000000000B
500000000000000"
 948 GO SUB rp: GO SUB door: GO
SUB door1: GO TO NX
 950 PRINT "You are in the priso
n's kitchen.There is a funny sme
ll coming  from a room to the s
outh.": LET r$="050005500000000
```

```
02000000000000000000000000"
 952 LET pos=0: GO SUB rp: PLOT
80,81: DRAW 0,64: DRAW 46,0: DRA
W 0,-64: PLOT 132,81: DRAW 0,30:
 DRAW 38,0: DRAW 0,-30: PLOT 81,
130
 953 DRAW 45,0: PLOT 80,111: DRA
W 45,0: PLOT 87,111: DRAW 0,3: D
RAW 10,0: DRAW 0,-3: PLOT 108,11
1: DRAW 0,3: DRAW 10,0: DRAW 0,-
3
 954 PLOT 152,110: DRAW 0,-30: C
IRCLE 143,96,2: CIRCLE 160,96,2:
 FOR q=0 TO 4: CIRCLE 88+(q*7),1
38,2: NEXT q: CIRCLE 119,96,2
 955 GO SUB doorr: GO TO NX
1000 PRINT "You are in the priso
n hall. You can see a sign on th
e wall whichgoes SE, it says 'PR
IVATE...NO  ENTRY TO UNAUTHORISE
D PERSONNEL'": LET r$="000000000
0001300125000000950055000000000"
1048 LET pos=0: GO TO 553
1050 PRINT "You have walked into
 a matter  transporter. There i
s a sign on the wall. You can't
get out.": LET r$="000000000000
00000000000000000000000000000"
1055 LET pos=20: GO SUB rp: PRIN
T AT 4,11;"Press 'T'";AT 6,14;"t
o";AT 8,11;"Transport": GO TO NX
1100 PRINT "You are in a small r
oom filled  with small pieces of
 broken up  stones. The only pas
sage out is east.": LET r$="0000
0000090000000000000000000000000
0000"
1148 GO SUB rp: GO SUB doorr: GO
 TO NX
1150 PRINT "You are in a passage
 which runs from north to south.
 You can seethe passage split up
 in the neardistance.": LET r$="
09000000000000014500000000009000
00000000"
1198 GO SUB passage: GO SUB padr
: GO TO NX
1200 PRINT "You are in the larde
r. The funnysmell was rotten dec
aying food. The only exit is nor
th to the  kitchen.": LET r$="0
9500000000000000000000000000000
0000000"
1202 IF food=0 THEN  PRINT AT 3,
1;"You can";AT 5,3;"see";AT 7,3;
"FOOD";AT 13,15;"□□"
1206 LET pos=23: GO SUB rp: GO S
UB door: GO TO NX
1250 PRINT "This is the prison s
toreroom. Itis empty unfortunate
ly for you. There is a posh look
ing room to the east.": LET r$="
10000000130000000000000000000000
00000000"
1298 GO SUB rp: GO SUB door: GO
SUB doorr: GO TO NX
1300 PRINT "You are in a very po
sh room. It is used by the Leade
r as his ownpersonal chamber. It
 is clean  and tidy, not like t
he rest of  the prison. A small
passage goesdown here.": LET r$=
"00000000000000000000000001250100
000001500"
1348 LET pos=0: GO SUB rp: GO SU
B door1: GO SUB hole: GO TO NX
1350 PRINT "You are at a junctio
n. Passages go east, NE and SW.
you can heara nasty roar from th
e east.": LET r$="0000105014000000000001550000000800000000000"
1398 LET pos=0: GO TO 1655
1400 PRINT "There is a huge cave
 monster in this cave and he is
roaring veryloudly. He is very h
ungry!": LET r$="000000000000000
000000001350000000000000"
1402 LET pos=27: GO SUB cave: GO
 SUB cdr1: GO TO NX
1450 PRINT "You are at a fork in
 the path.  It splits into 2 sep
arate paths.One goes SE, the oth
er SW.": LET r$="1150000000002250
000001750000000000000000000"
1498 LET pos=0: GO SUB 6200: GO
TO NX
1500 PRINT "You are standing bes
ide a small funny shaped toilet.
 The only  path out goes up.":
LET r$="00000000000000000000000000000000013000000"
1505 LET pos=29: GO SUB rp
1510 CIRCLE 135,116,2: PLOT 119,
80: DRAW 0,8: DRAW 16,0: DRAW 0,
-8: PLOT 119,88: DRAW -8,8: DRAW
 32,0: DRAW -8,-8: PLOT 115,96:
DRAW 0,16: PLOT 139,96: DRAW 0,1
6: PLOT 111,112: DRAW 32,0: DRAW
 0,8: DRAW -32,0: DRAW 0,-8: GO
TO NX
1550 PRINT "You are now on a nar
row twistingpath. It splits and
goes east,  north, down and NW.*
: LET r$="13500000160000000000000000000130000001850"
1560 GO SUB rp2: PLOT 64,56: DRA
W 72,62: DRAW 8,16: DRAW -8,8: D
RAW -40,8: DRAW -8,8: DRAW 8,8:
DRAW 30,8
1570 PLOT 152,56: DRAW 24,62: DR
AW -8,20: DRAW -8,8: DRAW -24,8:
 DRAW -32,6: DRAW 0,3: DRAW 30,1
2
1598 LET pos=0: GO TO NX
1600 PRINT "You are walking on a
 path from  east to west. The pa
th splits upto the east, and it
becomes verynarrow and twisting
to the west.": LET r$="00000000165000000000000015500000000000"
1648 GO SUB passage: GO SUB padr
: GO TO NX
1650 PRINT "The path splits into
 4 separate passages here. These
 go NE, SE, SW and NW. There is
a very deep shaft to the SW.": L
ET r$="000000800000021000000190016001050000000000"
1655 GO SUB 6200: PLOT 164,80: D
RAW  OVER 1;6,32: PLOT 165,80: D
RAW 26,16: DRAW 40,16: DRAW 24,2
2: PLOT 171,112: DRAW 24,8: DRAW
 16,0: DRAW 24,16: DRAW 20,8
1660 PLOT 103,143: DRAW -32,-16:
 DRAW -40,-8: DRAW -30,-24: PLOT
 105,143: DRAW  OVER 1;-10,3: PL
OT 100,146: DRAW -16,-8: DRAW -3
2,-8: DRAW -51,-4
1699 GO TO NX
1700 PRINT "You are in a corrido
r which goesdown quite steeply.
you can see a pit below you and
a corridor  to the east.": LET r
$="00000000175000000000000000000
00000001950"
1748 LET pos=0: GO SUB passage:
GO SUB hole: GO SUB padr: GO TO
NX
1750 PRINT "You are in an east w
est passage.There are only two p
aths going  away from here, they
 go east andwest.": LET r$="0000
0000145000000000000017000000000
00000"
1798 GO SUB passage: GO SUB padr
: GO TO NX
1800 PRINT "You are in a small l
ow room. Youcan see the canyon b
elow, but  you can't go back th
rough the  hole.": LET r$="0000
0000000000000002000000010000000
0000"
1848 LET pos=0: GO SUB rp: GO SU
B hole: GO TO NX
```

```
1850 PRINT "This is the old armo
urey. it hasbeen unused for cent
uries. The  room is thick with d
ust and it  smells foul. Paths g
o up and to the west only.": LET
 r$="0000000000000000000000000180
0000015500000"
1851 LET pos=36: IF laser=0 THEN
  PRINT AT 4,12;"You can";AT 6,1
3;"see a";AT 8,13;"LASER";AT 13,
15;"QR"
1898 GO SUB rp: GO SUB door1: GO
 TO NX
1900 PRINT "You are at the top o
f the lift  shaft. The lift has
disappeared back down the shaft,
 so you are unable to go back do
wn.": LET r$="0000160000000000000
0000000000000000000000"
1948 GO SUB rp: GO SUB doorr: GO
 TO NX
1950 PRINT "You are on the edge
of a large  pit. There is a brid
ge across itto a ledge on the ot
her side.": LET r$="000000000000
0000000000000000000017000000"
1951 LET pos=38
1955 GO SUB rp2: PLOT 0,111: DRA
W 255,0: PLOT 48,112: FOR g=1 TO
 5: DRAW 16,40: DRAW 16,-40: NEX
T g: PLOT 64,152: DRAW 128,0: DR
AW 16,8: DRAW -128,0: DRAW -16,-
8
1960 PLOT 208,160: DRAW 16,-40:
DRAW -16,-8: PLOT 223,120: DRAW
31,0: PLOT 0,120: DRAW 50,0: PLO
T 0,110: DRAW 72,0: DRAW -8,-16:
 DRAW 0,-8: DRAW -4,-16: DRAW 0,
-14: PLOT 72,110: DRAW 7,0: DRAW
 -8,-16: DRAW 0,-8: DRAW -4,-16:
 DRAW 0,-14
1965 PLOT 255,110: DRAW -64,0: D
RAW -4,-16: DRAW 0,-16: DRAW 4,-
16: DRAW -4,-6: GO TO NX
2000 PRINT "You are in a vast ca
nyon with a hole in the roof. If
 you go up  through this hole yo
u will not  be able to come back
 down.": LET r$="0000235000000000
0000000000000000018000000"
2048 LET pos=0: GO SUB cave: GO
TO NX
2050 PRINT "You are in a small c
ave with a  metal door set into
the wall.   The door has a small
 slot besideit.": LET r$="000000
0000000000000023500000000000000
00"
2098 LET pos=40: GO SUB cave: GO
 SUB cdrf: PLOT 103,112: DRAW 0,
12: DRAW 4,0: DRAW 0,-12: DRAW -
4,0: GO TO NX
2100 PRINT "You have now walked
into a smallmetal lift. The door
 has locked behind you. You can'
t get back  out. There are 2 but
tons on the wall labelled 'UP' a
nd 'DOWN'.": LET r$="00000000000
0000000000000000000019002400"
2102 LET pos=0: GO SUB rp: PLOT
96,80: DRAW 0,63: DRAW 56,0: DRA
W 0,-63: PLOT 124,80: DRAW 0,63
2103 PLOT 78,130: DRAW 10,0: DRA
W 0,-30: DRAW -10,0: DRAW 0,30:
CIRCLE 83,122,3: CIRCLE 83,108,3
2104 PRINT AT 6,3;"UP--->";AT 8,
3;"DOWN->": PLOT 88,152: DRAW 71
,0: DRAW 0,-4: DRAW -71,0: DRAW
0,4: GO SUB rp: GO TO NX
2150 PRINT "You are on the far s
ide of the  large pit. A bridge
crosses it  to the other side. Y
ou can see apath leading south.*
: LET r$="0000000000000000245000
0000000000000000000"
2151 LET pos=42: GO TO 1955
2200 BORDER 6: PAPER 6: CLS : PR
INT AT 15,0;"This room is extrem
ely bright.  It is lit up by ver
y powerful  lights. The room is
 empty and itlooks long deserted
.": LET r$="0000145022500000000
2450000000000002500"
2201 LET pos=43: IF torch=0 THEN
  PRINT AT 3,1;"You can";AT 5,2;
"see a";AT 7,2;"TORCH";AT 8,15;"
BB"
2248 GO SUB rp: GO SUB doorr: GO
 SUB hole: GO TO NX
2250 BORDER 4: PAPER 4: CLS : IN
K 0: PRINT AT 15,0;"You are on t
he edge of a massiveunderground
lake. You can see a bright light
 from the west and  you can hear
 splashing water in the east.":
LET r$="1200000023000000000000000
2200000000000000"
2251 LET pos=0: GO SUB rp2: PLOT
 0,55: DRAW 72,102: DRAW 112,0:
DRAW 70,-102
2255 LET w$="  ": PAPER 1: INK 1
: PRINT AT 4,15;w$;AT 5,13;w$;w$
;w$;AT 6,10;w$;w$;w$;w$;w$;w$(1)
;AT 7,8;w$;w$;w$;w$;w$;w$;w$;w$(
1);AT 8,6;w$;w$;w$;w$;w$;w$;w$;w
$;w$;w$;AT 9,5;w$;w$;w$;w$;w$;w$
;w$;w$;w$;w$;w$;AT 10,6;w$;w$;w$
;w$;w$;w$;w$;w$;w$;AT 11,7;w$;w$
;w$;w$;w$;w$;w$;w$;w$(1);AT 12,10
;w$;w$;w$;w$;w$;AT 13,13;w$;w$
2256 PAPER 4: INK 0: GO TO NX
2300 BORDER 3: PAPER 3: CLS : PR
INT AT 15,0;"You are on a river
jetty. There is a splashing rive
r running to the east. You can s
ee a large  lake to the west.":
 LET r$="0000000000000000000000000
0225000000000000"
2301 LET pos=45: GO SUB rp2: PLO
T 0,96: DRAW 79,0: DRAW 0,-39: F
OR q=1 TO 4: PRINT AT q+10,10; I
NK 1; PAPER 1;"
      ": NEXT q: PLOT 255,55: DRA
W 0,32: GO TO NX
2350 PRINT "You are in a small c
ave at one  end of a river. The
river goes  through a hole in th
e rock. You can see 4 paths lead
ing away.": LET r$="000020500000
0350000025500000200000000000"
2351 LET pos=46: GO SUB cave: GO
 TO NX
2400 PRINT "You are now at the b
ottom of thelift shaft. There is
 a dim lightcoming from the SW.
There is a  smell of oil from th
e SE.": LET r$="0000000000003350
0000275000000000190000000"
2448 GO SUB rp: GO SUB doorr: GO
 SUB door1: GO TO NX
2450 PRINT "You are in a narrow
passage thatruns from NE to SW.
There is a  bright light coming
from the NE.": LET r$="000022000
0000000000026000000000000000000"
2451 LET pos=0: GO SUB passage:
GO SUB padr: GO TO NX
2500 PRINT "You cannot believe y
our eyes !! This room is complet
ely filled  with sparkling new A
ndromedan  quonks...the local c
urrency.": LET r$="0000000000000
0000000000024500000000000000"
2501 LET pos=49: IF money=0 THEN
  PRINT AT 4,12;"You can";AT 6,1
4;"see";AT 8,13;"COINS";AT 12,10
;"NNNNNNNNNNNN";AT 13,7;"NNNNNNN
NNNNNNNNNNN"
2548 GO SUB rp: GO SUB door1: GO
 TO NX
2550 PRINT "You are in a small r
oom used by the prisoners for fo
rging guardssecurity passes.": L
ET r$="0000000000000000000000002
350000000000000"
2551 LET pos=50: IF pass=0 THEN
 PRINT AT 3,12;"You can";AT 5,14
;"see";AT 7,13;"a PASS";AT 12,14
;"GHI";AT 13,14;"JKL"
2598 GO SUB rp: GO SUB door1: GO
 TO NX
2600 PRINT "You are in a large b
are empty  canyon. A path leads
 SE to a bigpit. Two other paths
 go to the  NW and north.": LET
r$="1950000000002650000000000000
2450000000000"
2648 LET pos=0: GO SUB cave: GO
SUB cdrf: GO TO NX
2650 PRINT "You are on the edge
of a big pitfilled with very sha
rp spikes  tipped with deadly p
oison. The  pit is too big to ju
mp. It is  only safe to go the
way you came(NW).": LET r$="2700
2700270027002700270026000000
2700"
2651 LET pos=52: GO SUB cave: GO
 SUB hole: GO TO NX
```

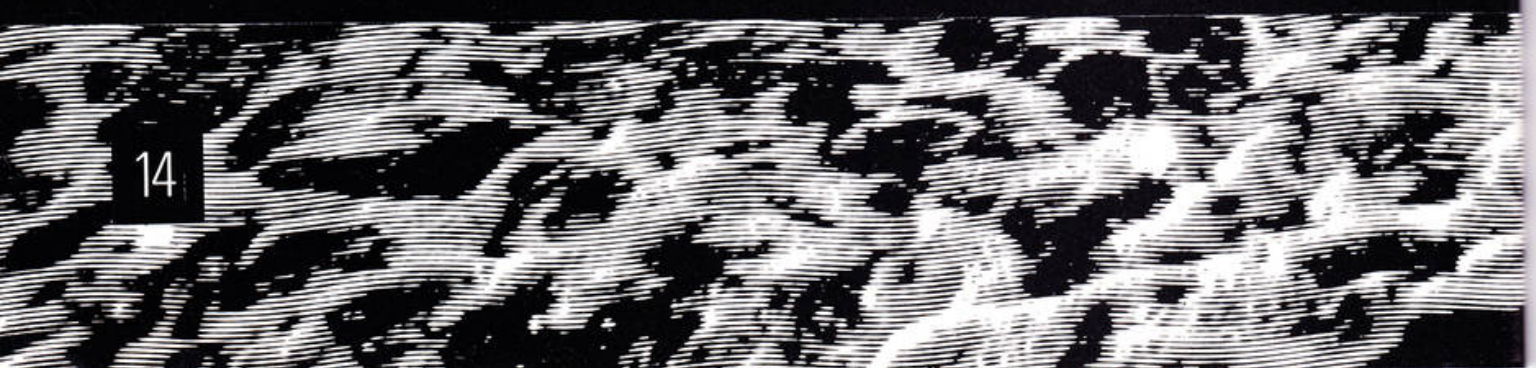
```
2700 CLS : LET a$="Oh dear...you
 are dead.": FOR f=1 TO 19: BEEP
 .02,(20-f): PRINT TAB INT (RND#
9);o : NEXT f
2710 PRINT  BRIGHT 1; FLASH 1;AT
 20,6;"ANOTHER GAME ? (Y/N)"
2720 IF INKEY$="Y" THEN  RUN
2730 IF INKEY$<>"N" THEN  GO TO
2720
2740 PRINT AT 20,6; BRIGHT 1; IN
VERSE 1; FLASH 1;"    YOU CHICKE
N!    ": STOP
2750 PRINT "You are now in a sma
ll dimly litcave. Paths lead dow
n, north andeast. To the east yo
u can see  the bottom of a lift
 shaft.": LET r$="210000240000
0000000000000000000000?000"
2752 LET pos=0: GO SUB cave: GO
SUB cdrf: GO SUB cdrr: GO SUB ho
le: GO TO NX
2800 PRINT "This room is used by
 the prison guards as a shooting
 range. You can see targets on a
ll of the  walls around you. Ap
art from thetargets, the room is
 empty.": LET r$="000000000000000
0000000000000000002750000"
2802 LET POS=55: GO SUB rp: FOR
q=1 TO 3: CIRCLE 96,130 q¥4: CIR
CLE 158,130,q¥4: CIRCLE 127,130,
q¥4: CIRCLE 112,98,q¥4: CIRCLE 1
43,98,q¥4: NEXT q: GO TO NX
2810 PRINT "You are in a magnifi
cent canyon.The only path leads
SE to what  looks like a dusty r
oom. In the distance you can sme
ll  rotting  human flesh and bone
s.": LET r$="000000000002900000
0000000000000000000000"
2851 LET pos=0: GO SUB cave: GO
TO NX
2900 PRINT "You are in a dusty a
nd decaying room. The smell of r
otting fleshis much stronger and
 it seems tobe coming from the n
orth.": LET r$="300000002950000
0000000000002850000000"
2948 GO SUB rp: GO SUB door: GO
SUB doorr: GO TO NX
2950 PRINT "This is an extremely
 small room filled with dust. Yo
u can just  see a small sign on
the wall.": LET r$="000000000000
0000000000290000000000000"
2951 GO SUB rp: GO SUB doorl: PR
INT AT 4,10;"COMBINATION";AT 6,1
4;"IS";AT 8,11;"'254637'": GO TO
 NX
3000 PRINT "The smell of rotting
 flesh is  very intense now. In
 the corner of this room you can
 see what isleft of a previous a
dventurer  who failed to escape
. Passages  out lead north, sout
h and SW.": LET r$="335000000000
0002900305000000000000000"
3048 GO SUB rp: GO SUB door: GO
TO NX
3050 PRINT "You are on a long na
rrow path.  The path leads NE an
d NW. There is a funny smell fro
m the NE.": LET r$="000030000000
0000000000000003100000000"
3098 GO SUB passage: GO SUB padr
: GO TO NX
3100 PRINT "You are on pebbled p
ath. It goessouth and SW.": LET
r$="000000000000000003050315000000
000000000000"
3148 GO TO 1560
3150 PRINT "You are walking on t
he pebbled  path. It splits into
 two paths. These go NE and west
. The path  to the west slopes u
p steeply.": LET r$="000031000000
0000000000000?2000000000000000"
3155 GO TO 1498
3200 PRINT "You are on a steep i
ncline whichleads up. There is a
 path with apebbled floor to the
 east. You  can see a cave upwar
ds.": LET r$="000000003150000000
0000000000000032500000"
3201 LET pos=0: GO SUB passage:
GO TO NX
3250 PRINT "You are in an old ro
om which hasbeen deserted for a
very long  time. A steep path l
eads down.": LET r$="000000000000
0000000000000000000003200"
3253 GO SUB rp: GO SUB hole: LET
 pos=64: IF ladder=0 THEN  PRINT
 AT 4,1;"You can";AT 6,2;"see a"
;AT 8,1;"LADDER";AT 10,13;"$$$$$
$"
3299 GO TO NX
3300 PRINT "You are in a small r
oom behind  the cave monster. Th
ere are onlytwo paths out of thi
s room. Theygo north and south.
You can see a split in a path to
 the south.": LET r$="340000000
0000016500000000000000000000"
3301 LET pos=0: GO SUB rp: GO SU
B door: GO TO NX
3350 PRINT "You are in a room fu
ll of old  disused machinery. I
t smells of oil. The machinery w
as used longago for working the
lift. Paths out go NE and NW.":
LET r$="000016500000000000000000
0000240000000000"
3398 GO SUB rp: GO SUB doorl: GO
 SUB doorr: GO TO NX
3400 PRINT "You are in a medium
large cave. There is a pit in th
e distance  to the SE, and other
 paths lead to the NE and south.
": LET r$="000010500000345033000
0000000000000000000"
3401 LET pos=0: GO SUB cave: GO
TO NX
3450 PRINT "You are on the edge
of a deep  pit. There is a larg
e column of glass leading to the
 bottom. It is too slippery to c
limb down.": LET r$="00000000000
0000000000000003400000000000"
3498 LET pos=68: GO SUB cave: GO
 SUB hole: GO TO NX
3500 PRINT "You are at the botto
m of the pitwith the glass wall
beside you. There is a passage l
eading away to the SW.": LET r$=
"000000000000000000000035500000000
034500000"
3501 LET pos=0: GO SUB cave: GO
SUB cdrl: GO TO NX
3550 PRINT "You are in a narrow
passage. Youcan hear voices from
 the east.  The only other path
out goes up.": LET r$="000000003
6000000000000000000000035000000"
3551 LET pos=0: GO SUB passage:
GO SUB padr: GO TO NX
3600 PRINT "You are in a small r
oom. You canhear the voices more
 loudly now.The voices are now c
oming from  the south. There is
also a largecombination lock on
the door.": LET r$="000000000000
0000000000003550000000000000"
```

```
3601 LET pos=71: GO SUB rp: GO S
UB doorr: GO TO NX
3650 PRINT "You are in a room fi
lled with a large group of guard
s. They needto see a security pa
ss in order to let you get past.
": LET r$="360000000000000000000
0000000000000000000"
3651 LET pos=72: GO SUB rp: GO S
UB door: GO SUB door1: GO TO NX
3700 PRINT #1;"Ouch...that's a w
all !          ": FOR L=1 TO 5:
BEEP .05,20: NEXT L: PAUSE 75: R
ETURN
3710 PRINT #1;"Sorry...but I don
't understand  ": BEEP .3,20: PA
USE 75: RETURN
3800 IF torch<>2 THEN  PRINT #1;
"You cannot see to go on": BEEP
.3,10: PAUSE 75: GO TO NX
3805 IF torch=2 THEN  LET RM=010
0: GO TO 9690
3810 IF key<>2 THEN  PRINT #1;"T
he door is locked": BEEP .3,10:
PAUSE 75: GO TO NX
3815 IF key=2 THEN  LET RM=0150:
 GO TO 9690
3820 IF torch=0 THEN  PRINT #1;"
You must take the torch first":
BEEP .3,10: PAUSE 75: GO TO NX
3825 IF torch=1 THEN  LET torch=
2: GO TO 251
3830 IF key=0 THEN  PRINT #1;"Yo
u haven't got the key": BEEP .3,
10: PAUSE 75: GO TO NX
3835 IF key=1 THEN  PRINT #1;"OK
...the door is open": LET key=2:
 BEEP .3,10: PAUSE 75: GO TO 101
3840 IF money=0 THEN  PRINT #1;"
You must get the coins first": B
EEP .3,10: PAUSE 75: GO TO NX
3845 IF money=1 THEN  PRINT #1;"
The guard accepts your bribe": B
EEP .3,10: LET money=2: PAUSE 75
: GO TO NX
3850 IF (pos<>3 OR disk<>0) THEN
  PRINT #1;"I don't see a disk h
ere": BEEP .3,10: PAUSE 75: GO T
O NX
3851 IF money<>2 THEN  PRINT #1;
"The guard won't let you have it
": BEEP .3,10: PAUSE 75: GO TO N
X
3855 PRINT #1;"OK...you have tak
en the disk": LET disk=1: BEEP .
3,10: PAUSE 75: GO TO NX
3860 IF (pos<>38 AND pos<>42) TH
EN  PRINT #1;"There is nothing t
o cross here": BEEP .3,10: PAUSE
 75: GO TO NX
3865 IF pos=38 THEN  LET RM=2150
: GO TO 9690
3870 IF pos=42 THEN  LET RM=1950
: GO TO 9690
3900 IF (pos<>23 OR food<>0) THE
N  PRINT #1;"There is no food he
re": BEEP .3,10: PAUSE 75: GO TO
 NX
3905 IF food=0 THEN  PRINT #1;"O
K...you have the food": LET food
=1: BEEP .3,10: PAUSE 75: GO TO
NX
3910 IF (pos<>43 OR torch<>0) TH
EN  PRINT #1;"The torch isn't he
re": BEEP .3,10: PAUSE 75: GO TO
 NX
3915 IF torch=0 THEN  PRINT #1;"
OK... you have the torch": LET to
rch=1: BEEP .3,10: PAUSE 75: GO
TO NX
3920 IF (pos<>49 OR money<>0) TH
EN  PRINT #1;"There are no coins
 here": BEEP .3,10: PAUSE 75: GO
 TO NX
3925 IF money=0 THEN  PRINT #1;"
OK...you have the coins": LET mo
ney=1: BEEP .3,10: PAUSE 75: GO
TO NX
3930 IF (pos<>14 OR key<>0) THEN
  PRINT #1;"I don't see a key he
re": BEEP .3,10: PAUSE 75: GO TO
 NX
3935 IF key=0 THEN  PRINT #1;"OK
...the key is yours": LET key=1:
 BEEP .3,10: PAUSE 75: GO TO NX
3940 IF (pos<>64 OR ladder<>0) T
HEN  PRINT #1;"There is no ladde
r here": BEEP .3,10: PAUSE 75: G
O TO NX
3945 IF ladder=0 THEN  PRINT #1;
"OK...you have the ladder": LET
ladder=1: BEEP .3,10: PAUSE 75:
GO TO NX
3950 IF (pos<>50 OR pass<>0) THE
N  PRINT #1;"There's no pass her
e": BEEP .3,10: PAUSE 75: GO TO
NX
3955 IF pass=0 THEN  PRINT #1;"O
K...you have the security pass":
 LET pass=1: BEEP .3,10: PAUSE 7
5: GO TO Nx
3960 IF (pos<>36 OR laser<>0) TH
EN  PRINT #1;"There's no laser h
ere": BEEP .3,10: PAUSE 75: GO T
O NX
3965 IF laser=0 THEN  PRINT #1;"
OK...you have the laser gun": LE
```



ANDROMEDA 3

```
T laser=1: BEEP  3,10: PAUSE 75:
 GO TO Nx
3970 CLS : PRINT "You are carryi
ng :-"
3971 IF disk<>0 THEN  PRINT : PR
INT "QR  (A METAL DISK)"
3972 IF key<>0 THEN  PRINT : PRI
NT "EE  (A SMALL KEY)"
3973 IF food<>0 THEN  PRINT : PR
INT "CD  (SOME FOOD)"
3974 IF torch<>0 THEN  PRINT : P
RINT "AB  A TORCH)"
3975 IF money<>0 THEN  PRINT : P
RINT "MNMNMNMN (A FEW COINS)"
3976 IF laser<>0 THEN  PRINT : P
RINT "QR  (A LASER GUN)"
3977 IF ladder<>0 THEN  PRINT :
PRINT "SSSS  (A LADDER)"
3978 IF pass<>0 THEN  PRINT : PR
INT "GHI  (A FORGED PASS)": PRIN
T "JKL"
3979 IF (disk=0 AND key=0 AND fo
od=0 AND torch=0 AND money=0 AND
 ladder=0 AND pass=0 AND laser=0
) THEN  PRINT : PRINT "NOTHING A
T ALL"
3980 PRINT AT 20,0; FLASH 1;"Pre
ss any key to carry on"
3985 IF INKEY$="" THEN  GO TO 39
85
3986 FLASH 0: CLS : LET RM=rmi:
GO TO 9695
4000 LET l=LEN a$: LET l=l-1
4001 IF a$(l)=" " THEN  GO TO 40
03
4002 LET l=l-1: GO TO 4001
4003 LET w$=a$(l+1 TO ): PRINT #
1;"You can't take the ";w$: BEEP
 .3,10: PAUSE 75: GO TO NX
4010 BORDER 2: PAPER 1: CLS : IN
K 7: LET z$="TRANSPORTING": LET
xy=0
4020 FOR j=1 TO 21: PRINT TAB xy
;z$: LET xy=xy+1: NEXT j
4030 FOR q=1 TO 15: FOR j=1 TO 3
0 STEP 3: BEEP .01,j: NEXT j: NE
XT q: LET RM=2200: GO TO 9690
4050 IF (pos<>45 AND pos<>46) TH
EN  PRINT #1;"There is no river
here": BEEP .3,10: PAUSE 75: GO
TO NX
4055 IF pos=45 THEN  LET RM=2350
: GO TO 9690
4060 IF pos=46 THEN  LET RM=2300
: GO TO 9690
4070 IF (pos<>38 AND pos<>42 AND
 pos<>52) THEN  PRINT #1;"There
is nothing to jump here": BEEP .
3,10: PAUSE 75: GO TO NX
4075 IF pos=52 THEN  LET RM=2700
: GO TO 9690
4080 PRINT #1;"The pit is too bi
g to jump": BEEP  3,10: PAUSE 75
: GO TO NX
4090 IF pos<>29 THEN  PRINT #1;"
I can't see the toilet": BEEP .3
,10: PAUSE 75: GO TO NX
4095 IF toilet=0 THEN  PRINT #1;
"AAAHHHH...what a relief !": LET
 toilet=1: BEEP .3,10: PAUSE 75:
 GO TO NX
4100 IF toilet=1 THEN  PRINT #1;
"You've just been !": BEEP .3,10
: PAUSE 75: GO TO NX
4110 IF food=0 THEN  PRINT #1;"Y
ou must take the food first": BE
EP .3,10: PAUSE 75: GO TO NX
4120 PRINT #1;"Yuk! you can't ea
t rotten food": BEEP .3,10: PAUS
E 75: GO TO NX
4130 LET l=LEN a$: LET l=l-1
4131 IF a$(l)=" " THEN  GO TO 41
33
4132 LET l=l-1: GO TO 4131
4133 LET w$=a$(l+1 TO ): PRINT #
1;"You can't eat the ";w$: BEEP
.3,10: PAUSE 75: GO TO NX
4150 IF pos<>27 THEN  PRINT #1;"
The monster is not here": BEEP .
3,10: PAUSE 75: GO TO NX
4155 IF food=0 THEN  GO TO 4110
4160 LET food=2: LET RM=3300: GO
 TO 9690
4170 IF laser=0 THEN  PRINT #1;"
You have to take the laser first
": BEEP .3,10: PAUSE 75: GO TO N
X
4175 IF pos<>55 THEN  PRINT #1;"
There's nothing to shoot at": BE
EP .3,10: PAUSE 75: GO TO NX
4180 LET RM=2850: GO TO 9690
4190 IF disk=0 THEN  PRINT #1;"y
ou need to take the disk first":
 BEEP .3,10: PAUSE 75: GO TO NX
4195 IF pos<>40 THEN  PRINT #1;"
There's nowhere to put it in": B
EEP .3,10: PAUSE 75: GO TO NX
4200 LET RM=2100: GO TO 9690
4210 IF pass=0 THEN  PRINT #1;"Y
ou need to take the pass first":
 BEEP .3,10: PAUSE 75: GO TO NX
4215 IF pos<>72 THEN  PRINT #1;"
There's no-one to show it to": B
EEP .3,10: PAUSE 75: GO TO NX
4220 GO TO 4500
4300 IF ladder=0 THEN  PRINT #1;
"You must take the ladder first"
: BEEP .3,10: PAUSE 75: GO TO NX
4305 IF pos<>68 THEN  PRINT #1;"
There's nowhere to use it": BEEP
 .3,10: PAUSE 75: GO TO NX
4310 PRINT #1;"OK...the ladder i
s in the pit": LET ladder=2: BEE
P .3,10: PAUSE 75: GO TO NX
4320 IF pos<>71 THEN  PRINT #1;"
Nothing happens": BEEP .3,10: PA
USE 75: GO TO NX
4321 PRINT #1;"Enter the number
or S to stop": BEEP .3,10: PAUSE
 75
4322 INPUT a$
4323 IF a$="S" THEN  PRINT #1;"O
K...back to normal": BEEP .3,10:
 PAUSE 75: GO TO NX
4325 IF a$<>"254637" THEN  PRINT
 #1;"INCORRECT NUMBER": BEEP .3,
10: PAUSE 75: GO TO 4322
4330 LET RM=3650: GO TO 9690
4350 IF ladder<>2 THEN  PRINT #1
;"The glass is too slippery": BE
EP .3,10: PAUSE 75: GO TO NX
4355 LET RM=3500: GO TO 9690
4500 BORDER 0: PAPER 0: CLS : IN
K 3
4510 FOR Q=1 TO 21: PRINT TAB Q;
 BRIGHT 1; FLASH 1;"WELL DONE":
NEXT Q
4520 FOR H=1 TO 5: FOR K=-10 TO
20 STEP 3: BEEP .03,K: BEEP .03,
(20-K): NEXT K: NEXT H
4530 BORDER 3: PAPER 1: CLS : IN
K 7: PRINT  BRIGHT 0;"Well done,
 you have escaped fromthe underg.
round prison.": PRINT : PRINT "Y
ou steal a spaceship and returnt
o Earth just in time to stop  t
he war."
4540 PRINT : PRINT  BRIGHT 1; FL
ASH 1;" CONGRATULATIONS,  YOU HA
VE WON ": PRINT  BRIGHT 1; FLASH
 1;AT 15,11;"GAME OVER"
4550 STOP
5000 GO SUB rp2: PLOT 72,80: DRA
W 104,0: DRAW 0,79: DRAW -104,0:
 DRAW 0,-79: PLOT 0,175: DRAW 71
,-16: PLOT 177,159: DRAW 77,16:
PLOT 0,56: DRAW 71,24: PLOT 255,
56: DRAW -78,24: RETURN
5100 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
 PLOT 0,55: DRAW 255,0: RETURN
5200 PLOT 103,81: DRAW 0,56: DRA
W 44,0: DRAW 0,-56: PLOT 103,80:
 DRAW 32,8: DRAW 0,48: CIRCLE 12
7,112,2: RETURN
5300 PLOT 239,62: DRAW 0,88: DRA
W -48,-8: DRAW 0,-65: CIRCLE 231
,103,2: RETURN
5400 PLOT 16,62: DRAW 0,88: DRAW
 48,-8: DRAW 0,-65: CIRCLE 24,10
3,2: RETURN
5500 PLOT 72,64: DRAW 110,0: DRA
W -24,8: DRAW -64,0: DRAW -24,-8
: PLOT 94,72: DRAW 0,-8: PLOT 15
8,72: DRAW 0,-8: RETURN
5600 GO SUB rp2: PLOT 0,55: DRAW
 80,40: DRAW 94,0: DRAW 80,-40:
PLOT 0,55: DRAW 8,16: DRAW 8,24:
 DRAW -8,24: DRAW 0,8: DRAW 16,2
4: DRAW 16,8: DRAW 24,8: DRAW 24
,-8: DRAW 40,0
5650 DRAW 24,8: DRAW 16,-8: DRAW
 16,8: DRAW 24,-8: DRAW 24,0: DR
AW 16,-16: DRAW -16,-32: DRAW 8,
-16: DRAW 8,-8: DRAW -8,-16: DRA
W 15,-16
5660 PLOT 80,96: DRAW 8,16: DRAW
 8,24: DRAW -8,23: PLOT 172,96:
DRAW -16,16: DRAW 8,16: DRAW -8,
8: DRAW -8,16: DRAW 7,14: RETURN

5700 PLOT 31,72: DRAW 0,48: DRAW
 32,16: DRAW 0,-48: CIRCLE 37,10
0,2: RETURN
5800 PLOT 191,88: DRAW 0,48: DRA
W 32,-16: DRAW 0,-48: CIRCLE 216
,100,2: RETURN
5900 PLOT 112,96: DRAW 0,40: DRA
W 16,8: DRAW 16,-8: DRAW 0,-40:
CIRCLE 136,115,2: RETURN
```



```
6000 GO SUB rp2: PLOT 104,96: DR
AW 46,0: DRAW 0,38: DRAW -46,0:
DRAW 0,-38: DRAW -104,-40: PLOT
151,96: DRAW 102,-40: PLOT 151,1
35: DRAW 102,38: PLOT 104,135: D
RAW -104,38: RETURN
6100 PLOT 120,96: DRAW 0,24: DRA
W 14,0: DRAW 0,-24: RETURN
6200 GO SUB rp2: PLOT 48,56: DRA
W 102,80: DRAW -48,8: DRAW -24,8
: DRAW -40,22: PLOT 159,56: DRAW
 16,80: DRAW 24,16: DRAW 56,16
6210 PLOT 50,175: DRAW 16,-8: DR
AW 40,-12: DRAW 55,-4: DRAW 24,8
: DRAW 70,12: RETURN
9500 PRINT AT 21,0;"What now ?":
 PLOT 0,0: DRAW 0,8: PLOT 0,0: D
RAW 80,0: INPUT a$
9501 LET rmi=rm
9502 IF a$="Z" THEN  COPY : GO T
O NX
9505 LET r=0: LET r=(1 AND a$="N
")+(2 AND a$="NE")+(3 AND a$="E"
)+(4 AND a$="SE")+(5 AND a$="S")
+(6 AND a$="SW")+(7 AND a$="W")+
(8 AND a$="NW")+(9 AND a$="U")+(
10 AND a$="D")
9510 IF a$="E" AND pos=1 THEN  G
O TO 3810
9520 IF a$="N" AND pos=4 THEN  G
O TO 3800
9530 IF a$="D" AND pos=68 THEN
GO TO 4350
9660 IF r=0 THEN  GO TO 9700
9670 LET RM=VAL r$(r*4-3 TO r*4)
9680 IF RM=0 THEN  GO SUB 3700:
GO TO NX
9690 LET paper=INT (RND*6)+1
9693 BORDER paper: PAPER paper:
INK 9: CLS
9695 PRINT AT 15,0;: GO TO RM
9710 IF a$="" THEN  GO SUB 3710:
 GO TO NX
9715 IF (a$="LIGHT THE TORCH" OR
 a$="LIGHT TORCH") THEN  GO TO 3
820
9720 IF (a$="UNLOCK THE DOOR" OR
 a$="UNLOCK DOOR" OR a$="USE THE
 KEY" OR a$="USE KEY" OR a$="OPE
N THE DOOR" OR a$="OPEN DOOR") T
HEN  GO TO 3830
9725 IF (a$="CROSS THE BRIDGE" O
R a$="CROSS BRIDGE" OR a$="CROSS
 OVER THE BRIDGE" OR a$="CROSS O
VER BRIDGE") THEN  GO TO 3860
9730 IF  a$="JUMP THE PIT" OR a$
="JUMP PIT"  OR a$="JUMP OVER THE
 PIT" OR a$="JUMP OVER PIT" OR a
$="JUMP ACROSS THE PIT" OR a$="J
UMP ACROSS PIT") THEN  GO TO 407
0
9735 IF (a$="GET THE DISK" OR a$
="GET DISK" OR a$="TAKE THE DISK
" OR a$="TAKE DISK") THEN  GO TO
 3850
9740 IF (a$="BRIBE THE GUARD" OR
 a$="BRIBE GUARD" OR a$="GIVE TH
E COINS TO THE GUARD" OR a$="GIV
E COINS TO THE GUARD" OR a$="GIV
E THE COINS TO GUARD"  THEN  GO
TO 3840
9745 IF (a$="GET THE KEY" OR a$=
"GET KEY" OR a$="TAKE THE KEY" O
R a$="TAKE KEY") THEN  GO TO 393
0
9750 IF (a$="GET THE FOOD" OR a$
="GET FOOD" OR a$="TAKE THE FOOD
" OR a$="TAKE FOOD"  THEN  GO
 3900
9755 IF (a$="GET THE TORCH" OR a
$="GET TORCH" OR a$="TAKE THE TO
RCH" OR a$="TAKE TORCH"  THEN  G
O TO 3910
9760 IF (a$="GET THE COINS" OR a
$="GET COINS" OR a$="TAKE THE CO
INS" OR a$="TAKE COINS") THEN  G
O TO 3920
9765 IF (a$="GET THE LADDER" OR
a$="GET LADDER" OR a$="TAKE THE
LADDER" OR a$="TAKE LADDER"  THE
N  GO TO 3940
9770 IF (a$="GET THE PASS" OR a$
="GET PASS" OR a$="TAKE THE PASS
" OR a$="TAKE PASS") THEN  GO TO
 3950
9780 IF (a$="GET THE LASER" OR a
$="GET LASER" OR a$="TAKE THE LA
SER" OR a$="TAKE LASER") THEN  G
O TO 3960
9790 IF (a$="LIST" OR a$="L") TH
EN  GO TO 3970
9800 IF (a$="T" AND pos=20) THEN
  GO TO 4010
9810 IF (a$="SWIM THE RIVER" OR
a$="SWIM RIVER" OR a$="SWIM IN T
HE RIVER" OR a$="SWIM IN RIVER"
OR a$="SWIM") THEN  GO TO 4050
9820 IF (a$="GO TO THE TOILET" O
R a$="GO TOILET" OR a$="USE THE
TOILET" OR a$="USE TOILET" OR a$
="GO TO TOILET") THEN  GO TO 409
0
9830 IF (a$="EAT THE FOOD" OR a$
="EAT FOOD") THEN  GO TO 4110
9840 IF (a$="FEED THE CAVE MONST
ER" OR a$="FEED THE MONSTER" OR
a$="FEED MONSTER" OR a$="FEED CA
VE MONSTER" OR a$="GIVE THE MONS
TER THE FOOD" OR a$="GIVE MONSTE
R FOOD" OR a$="GIVE THE MONSTER
FOOD" OR a$="GIVE MONSTER THE FO
OD" OR a$="GIVE THE CAVE MONSTER
 THE FOOD" OR a$="GIVE CAVE MONS
TER THE FOOD" OR a$="GIVE CAVE M
ONSTER FOOD") THEN  GO TO 4150
9850 IF (a$="FIRE THE LASER" OR
a$="FIRE LASER" OR a$="SHOOT THE
 LASER" OR a$="SHOOT LASER") THE
N  GO TO 4170
9860 IF (a$="USE THE DISK" OR a$
="USE DISK" OR a$="PUT THE DISK
IN THE SLOT" OR a$="PUT DISK IN
THE SLOT" OR a$="PUT DISK IN SLO
T") THEN  GO TO 4190
9870 IF (a$="SHOW THE PASS" OR a
$="SHOW PASS" OR a$="SHOW THE GU
ARDS THE PASS" OR a$="SHOW GUARD
S THE PASS") THEN  GO TO 4210
9880 IF (a$="USE THE COMBINATION
" OR a$="USE COMBINATION") THEN
 GO TO 4320
9890 IF (a$="USE THE LADDER" OR
a$="USE LADDER") THEN  GO TO 430
0
9900 IF LEN a$<4 THEN  GO TO 993
0
9910 IF (a$(1 TO 4)="GET " OR a$
(1 TO 4)="TAKE") THEN  GO TO 4E3
9920 IF a$(1 TO 4)="EAT " THEN
GO TO 4130
9930 GO SUB 3710: GO TO NX
```

MATHS PACKAGE

BY COLIN CARRUTHERS

An end to tedious calculations in this program which enables the computer to solve a variety of mathematical puzzles.

Are you bored solving systems of equations with 5 unknowns? Want a hand to invert a couple of matrices? Shouldn't you check that difficult integration problem you've just spent four hours doing? Well, the computer's ability to plough through tedious calculations at high speed is used in this program to provide a useful maths package.

This program will run on both the 16K and 48K Spectrums, but since it is written entirely in BASIC it should be quite a simple task to implement the package on other micros. As far as possible, each part of the program has been made self-contained enabling the individual to just type in the routines required. However, the 'Matrix Operations' section demands that the 'System of Equations' routines be present — this is due to the fact that matrix inversion and solving systems of equations can be done by similar techniques and therefore have common program blocks. In any case, the program must have the 'Menu', 'Input a number' and 'Hit any key' routines — see figure 1, "Program Breakdown".

There now follows a brief description of each part of the program and examples of the kind of mathematical problem, they solve. these examples can be used to check that the routine have been typed in correctly.

System of Equations

Solved using Gaussian Elimination, each problem can have a maximum of five equations and five unknowns. The coefficients are held within a two-dimensional array (called 'a'). The user is prompted for each coefficient of x in turn, with the whole array of values shown on the screen at all times to enable checking.

Ex (n = 3)

$$x_1 + 3x_2 - 4x_3 = -11$$
$$2x_1 - x_2 + 3x_3 = 10$$
$$4x_1 + x_2 - 2x_3 = 3$$

This has the solution $x_1 = 2$, $x_2 = -3$ and $x_3 = 1$.

Quadratic Equation

The roots are found using the classic formula:

$$x = \frac{-b \sqrt{b^2-4ac}}{\sqrt{2a}}$$

This routine allows for both real and imaginary roots.

Ex 1 $x^2-3x+2=0$ gives x=1 or 2
Ex 2 $x^2-6x+10=0$ gives $x=3\pm i$
Ex 3 $x^2-6x+9=0$ gives x=3 (double root)

Equation of Third Degree

This routine gives the roots of a polynomial with a term in x^3. Again, imaginary roots are catered for, giving 4 types of possible solution.

Ex1 $x^3-6x^2+11x-6=0$ gives x=1,2,3
Ex2 $x^3+3x-1=0$ gives x=1,1,1
Ex3 $-x^2-9x^2+81x+729=0$ gives x=9,−9,−9
Ex4 $x^3-5x^2+7x+13=0$ gives $x=-1,3\pm 2i$

Matrix Operations

The determinant of the given square matrix is calculated and displayed. Assuming that this is non-zero, the inverse is computed using Gaussian Elimination. (A matrix with zero determinant has no inverse). The main 'invert' routine is the same as that for the 'System of Equations'.

Ex (n=3) $\begin{bmatrix} 3 & 1 & 2 \\ 2 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}$ has determinant 1 and inverse $\begin{bmatrix} 1 & 1 & -2 \\ -2 & -1 & 4 \\ 0 & -1 & 1 \end{bmatrix}$

Note that only real matrix elements are allowed.

Simpson's rule

The function entered must be a valid expression in 'x', for example 'y=3x+2' must be entered as:

y=3 *x+2

Also, functions such as SIN, TAB or LN must be entered as single-stroke key words. Any invalid expression typed in response to the prompt will result in an error at Line 7100, statement 3. If this should happen, simply type:

GOTO 7000

and re-type the expression correctly.

Ex y=4+x
lower x=0
upper x=4
samples=10

As can be seen by looking at figure 2, the value of the integral (ie the shaded part of the graph) should be 8.

Type in the program as it is listed and check the operation of each segment using the given examples. they have been chosen to test all parts of the program and should make any typing errors apparent. When all is well, type CLEAR and save to tape or Microdrive to auto-run from line 100.

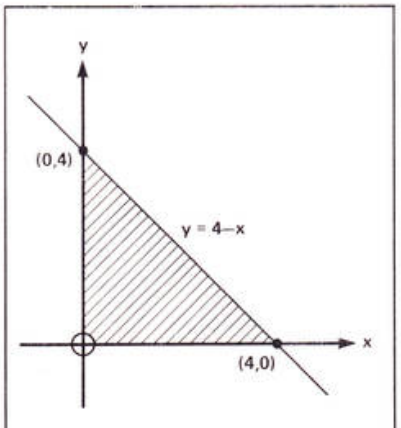


Figure 1.

Program Breakdown.

500-580	:'Input a number' routine. Extensive use is made of this routine, so it is placed near the beginning to speed execution.
1000-1180	:Menu.
2000-2970	:System of Equations.
3000-3500	:Quadratic Equation.
4000-4500	:Equation of 3rd degree.
6000-6650	:Matrix Determinant and Inverse.
7000-7500	:Simpson's Rule.
9000-9030	:'Hit any Key to Return'.

```
100 GO TO 1000
500 LET a$="": LET dps=0
505 IF integer THEN LET dps=1
510 PRINT AT x,y;a$; INK 4;"
515 IF INKEY$<>"" THEN GO TO 5
15
520 LET z$=INKEY$: IF z$="" THE
N GO TO 520
525 BEEP .02,34.4: IF LEN a$>=1
engtH THEN GO TO 560
530 IF z$="." AND NOT dps THEN
LET a$=a$+z$: LET dps=1: GO TO
510
540 IF z$="-" AND a$="" THEN L
ET a$=z$: GO TO 510
550 IF z$>="0" AND z$<="9" THEN
LET a$=a$+z$: GO TO 510
560 IF CODE z$=13 AND a$<>"" AN
D a$<>"-" THEN PRINT AT x,y;a$;
" ": RETURN
565 IF CODE z$=12 AND LEN a$>=1
THEN IF a$(LEN a$)="." THEN L
ET dps=0
570 IF CODE z$=12 AND LEN a$>=1
THEN LET a$=a$( TO LEN a$-1):
GO TO 510
580 BEEP .5, 10: GO TO 510
1000 REM menu
1010 BORDER 0: INK 7: PAPER 0
1020 CLS : POKE 23609,30
1030 PRINT AT 3,3;"1 System of
Equations";AT 5,3;"2 Quadratic
Equation";AT 7,3;"3 Equation of
3rd Degree";AT 9,3;"4 Matrix O
perations";AT 11,3;"5 Simpson's
Method"
1035 PRINT AT 13,3;"0 Quit"
1040 PRINT AT 21,3;"Enter Number
->"
1050 LET x=21: LET y=20: LET int
eger=1: LET length=1
1060 GO SUB 500: IF a$>"5" THEN
BEEP .5,-10: GO TO 1060
1065 IF a$="0" THEN STOP
1070 IF a$="1" THEN GO SUB 2000
1075 IF a$="2" THEN GO SUB 3000
1080 IF a$="3" THEN GO SUB 4000
1085 IF a$="4" THEN GO SUB 6000
1090 IF a$="5" THEN GO SUB 7000
1100 RUN
2000 REM simultaneous
2005 LET t$="System of Equations
"
2010 LET matrix=0: CLS : PRINT A
T 1,3;t$;AT 5,3;"a(11)*x1 + a(12
)*x2 .. = b1";AT 6,3;"a(21)*x1 +
a(22)*x2 .. = b2"
2019 PRINT AT 7,3;"  .  .  .
.  .  .
2020 PRINT AT 8,3;"  .  .  .
.  .  .";AT 9,3;"a(n1)*x1 +
a(n2)*x2 .. = bn"
2030 PRINT AT 15,3;"Enter Order
n (2-5) -> "
2040 LET x=15: LET y=26: LET int
eger=1: LET length=1: GO SUB 500
2045 IF a$<"2" OR a$>"5" THEN B
EEP .5,-10: GO TO 2040
2050 LET n=VAL a$
2052 LET max=n+1: IF matrix THEN
 LET max=2*n
2053 DIM a(n,max)
2055 CLS : PRINT AT 0,3;t$
2057 LET y=12: LET integer=0: LE
T length=5
2058 IF NOT matrix THEN FOR j=1
TO n: PRINT AT 3,6*(j-1);"x";j;
NEXT j
2060 FOR i=1 TO n: FOR j=1 TO n
2062 PRINT AT 2*i+3,6*(j-1);"a("
;i;j;")"
2064 NEXT j: IF NOT matrix THEN
PRINT AT 2*i+3,6*(j-1);"b";i
2066 NEXT i
2068 FOR i=1 TO n
2072 LET upper=n+1: IF matrix TH
EN LET upper=n
2080 FOR j=1 TO upper
2090 IF j<=n THEN PRINT AT 15+j
,3;"a(";i;j;") = ";: GO TO 2100
2095 PRINT AT 15+j,3;"  b";i;"
= "
2100 LET x=15+j: GO SUB 500
2105 LET a(i,j)=VAL a$
2107 NEXT j
2109 FOR c=15 TO 21: PRINT AT c,
0;"
  ";: NEXT c
2110 GO SUB 2800: NEXT i
2120 IF matrix THEN GO SUB 6500
2130 IF matrix THEN IF det=0 TH
EN GO TO 9000
2200 FOR i=1 TO n
2210   LET piv=a(i,i)
2215   IF piv=0 THEN GO TO 2900
2220   FOR j=1 TO max
2230     LET a(i,j)=a(i,j)/piv
2240   NEXT j
2250   FOR k=i+1 TO n
2255     LET piv=a(k,i)
2260     FOR j=i TO max
2270       LET a(k,j)=a(k,j)-pi
v*a(i,j)
2280     NEXT j
2290   NEXT k
2300 NEXT i
2310 FOR i=n-1 TO 1 STEP -1
2320   FOR j=i+1 TO n
2330     LET piv=a(i,j)
2340     FOR k=i+1 TO max
2350       LET a(i,k)=a(i,k)-piv
*a(j,k)
2360     NEXT k
2370   NEXT j
2380 NEXT i
2390 IF NOT matrix THEN FOR i=1
TO n: PRINT AT 15+i,3;"x";i;" =
";a(i,n+1): NEXT i: GO TO 9000
2400 FOR i=1 TO n: FOR j=1 TO n
2410 LET a(i,j)=a(i,j+n)
2420 NEXT j: GO SUB 2800: NEXT i
2425 PRINT : PRINT "Inverse"
2430 GO TO 9000
2800 PRINT AT 2*i+3,0;"
                "
2810 FOR k=1 TO upper: IF ABS a(
i,k)<0.001 THEN LET a(i,k)=0
2820 LET n$=STR$ a(i,k)+"  ":
LET n$=n$( TO 5): IF n$(1)<>"-"
THEN LET n$=" "+n$( TO 4)
2830 PRINT AT 2*i+3,6*(k-1);n$;
NEXT k: RETURN
2900 LET new=0
2905 IF matrix THEN GO TO 2940
2910 FOR p=i+1 TO n
2920 IF a(p,i)<>0 THEN LET new=
p
2930 NEXT p
2940 IF new=0 THEN PRINT AT 16,
3;"No Solution": GO TO 9000
2960 FOR m=1 TO n+1: LET temp=a(
i,m): LET a(i,m)=a(new,m): LET a
(new,m)=temp: NEXT m
2965 LET piv=a(i,i)
2970 GO TO 2220
```

```
3000 REM quadratic
3010 CLS : PRINT AT 1,3;"Quadrat
ic Equation";AT 5,3;"Axx + Bx +
C = 0"
3020 LET x=9: LET y=7: LET integ
er=0: LET length=6: PRINT AT 9,3
;"A = ": GO SUB 500: LET a=VAL a
$
3024 IF a=0 THEN  BEEP .5,-10: G
O TO 3020
3025 LET x=11: PRINT AT 11,3;"B
= ": GO SUB 500: LET b=VAL a$
3030 LET x=13: PRINT AT 13,3;"C
= ": GO SUB 500: LET c=VAL a$
3040 LET b=-b/2/a: LET d=b*b-c/a
3050 IF d=0 THEN  GO TO 3100
3055 IF d>0 THEN  GO TO 3200
3060 LET y=SQR (-d)
3070 PRINT AT 17,3;"Real part: "
;b;AT 19,3;"Imaginary: +/- ";y:
GO TO 3500
3100 PRINT AT 17,3;"Equal roots:
 ";b
3110 GO TO 3500
3200 PRINT AT 17,3;"Real: ";b+SQ
R d;AT 18,3;"Real: ";b-SQR d
3500 GO TO 9000
4000 REM 3rd degree
4010 CLS : PRINT AT 1,3;"Equatio
n of 3rd Degree";AT 5,3;"Axxx +
Bxx + Cx + D = 0"
4020 LET x=9: LET y=7: LET integ
er=0: LET length=6: PRINT AT 9,3
;"A = ": GO SUB 500: LET a=VAL a
$: IF a=0 THEN  BEEP .5,-10: GO
TO 4020
4030 LET x=10: PRINT AT x,3;"B =
": GO SUB 500: LET b=VAL a$
4040 LET x=11: PRINT AT x,3;"C =
": GO SUB 500: LET c=VAL a$
4050 LET x=12: PRINT AT x,3;"D =
": GO SUB 500: LET d=VAL a$
4060 LET b=b/a/3: LET c=c/a: LET
 d=d/a
4070 LET a=c/3-b*b
4080 LET e=d-b*c+2*b*b*b
4090 LET h=4*a*a*a+e*e
4100 IF ABS h<10^-8 THEN  GO TO
4300
4105 IF h>0 THEN  GO TO 4200
4110 LET f=2*SQR (-a)
4120 LET g=ACS (e/(2*a*SQR (-a))
)/3
4125 LET a=ASN 1: LET e=ASN .5
4130 LET c=f*SIN (a-g): LET d=-f
*SIN (e+g)
4140 LET i=-f*SIN (e-g)
4150 LET c=c-b: LET d=d-b: LET i
=i-b
4160 PRINT AT 16,3;"Real: ";c;AT
 17,3;"Real: ";d;AT 18,3;"Real:
";i
4170 GO TO 9000
4200 LET h=SQR h: LET f=.5*(h-e)
: LET g=-.5*(h+e): LET h=1/3
4210 LET f=ABS f^h*SGN f
4220 LET g=ABS g^h*SGN g: LET h=
0.5*SQR 3
4230 PRINT AT 15,3;"Real root: "
;f+g-b
4240 PRINT AT 17,3;"Real part: "
;-.5*(f+g)-b
4250 PRINT AT 18,3;"Imaginary: +
/- ";h*ABS (f-g): GO TO 9000
4300 IF ABS a<10^-8 THEN  PRINT
AT 16,3;"Triple: ";-b: GO TO 900
0
4310 LET f=-ABS (.5*e)^(1/3)*SGN
 e
4320 PRINT AT 16,3;"Real:
";2*f-b
4330 PRINT AT 17,3;"Equal root:
";-f-b
4500 GO TO 9000
6000 REM Matrix
6010 CLS : LET matrix=1: LET t$=
"Matrix Operations"
6020 PRINT AT 1,3;t$;AT 8,3;"a(1
1)  a(12) .. a(1n)";AT 9,3;"a(21
)  a(22) .. a(2n)";AT 10,3;"  .
  .  .";AT 11,3;"a(n1)
a(n2) .. a(nn)"
6025 PRINT AT 4,3;"Inverse and D
eterminant";AT 5,3;"of nxn Matri
x"
6030 GO TO 2030
6500 REM det
6505 FOR i=1 TO n: FOR j=1 TO n
6510 LET a(i,j+n)=a(i,j)
6512 NEXT j: NEXT i
6514 LET det=1
6516 FOR m=n TO 2 STEP -1
6518 LET p=a(m,m+n)
6520 IF p=0 THEN  GO TO 6600
6530 FOR i=1 TO m-1
6540 LET q=a(i,m+n)/p
6550 FOR j=1 TO m
6560 LET a(i,j+n)=a(i,j+n)-q*a(m
,j+n)
6570 NEXT j
6580 NEXT i
6590 NEXT m
6595 FOR i=1 TO n: LET det=det*a
(i,i+n): FOR j=1 TO n: LET a(i,j
+n)=0: NEXT j: LET a(i,i+n)=1: N
EXT i
6597 PRINT AT 17,3;"Determinant
= ";det
6598 IF det=0 THEN  PRINT AT 18,
3;"-> No Inverse"
6599 RETURN
6600 LET new=0: FOR f=1 TO m-1
6610 IF a(f,m+n)<>0 THEN  LET ne
w=f
6620 NEXT f
6630 IF new=0 THEN  LET det=0: G
O TO 6595
6640 FOR f=1 TO m: LET a(m,f+n)=
a(m,f+n)+a(new,f+n): NEXT f
6645 LET p=a(new,m+n)
6650 GO TO 6530
7000 REM simpson's
7010 CLS : PRINT AT 1,3;"Simpson
's Method -";AT 3,3;"Approximate
integral"
7020 PRINT AT 10,3;"Enter a funt
ion in x ";AT 12,3;"e.g.    y =
3*x+2"
7030 INPUT "      y = "; LINE f$
7040 PRINT AT 10,3;"
          ";AT 12,3;"
    ";AT 8,9;"y = ";f$
7050 PRINT AT 10,3;"lower x = ":
LET integer=0: LET x=10: LET y=
13: LET length=8: GO SUB 500: LE
T d=VAL a$
7060 PRINT AT 12,3;"upper x = ":
LET x=12: GO SUB 500: LET e=VAL
a$
7065 IF e<=d THEN  BEEP .5,-10:
GO TO 7060
7070 PRINT AT 14,3;"samples = ":
LET x=14: GO SUB 500: LET p=VAL
a$
7080 IF p<=0 THEN  BEEP .5,-10:
GO TO 7070
7090 LET h=(e-d)/2/p
7100 LET a=0: LET x=d: LET y=VAL
 f$
7110 LET a=a+y: LET x=x+h: LET y
=VAL f$
7120 LET a=4*y+a: LET x=x+h: LET
 y=VAL f$
7130 LET a=y+a: LET p=p-1
7140 IF p<>0 THEN  GO TO 7110
7150 LET c=a*h/3
7160 PRINT AT 17,2;"Integral = "
;c
7500 GO TO 9000
9000 LET a$="": IF INKEY$<>"" TH
EN  GO TO 9000
9010 PRINT AT 21,3; INK 4; INVER
SE 1;"HIT ANY KEY TO RETURN"
9020 IF INKEY$="" THEN  GO TO 90
20
9030 BEEP .02,34.4: RETURN
```

MATHS PACKAGE

CITY BOMBER

BY BARRY McGAW

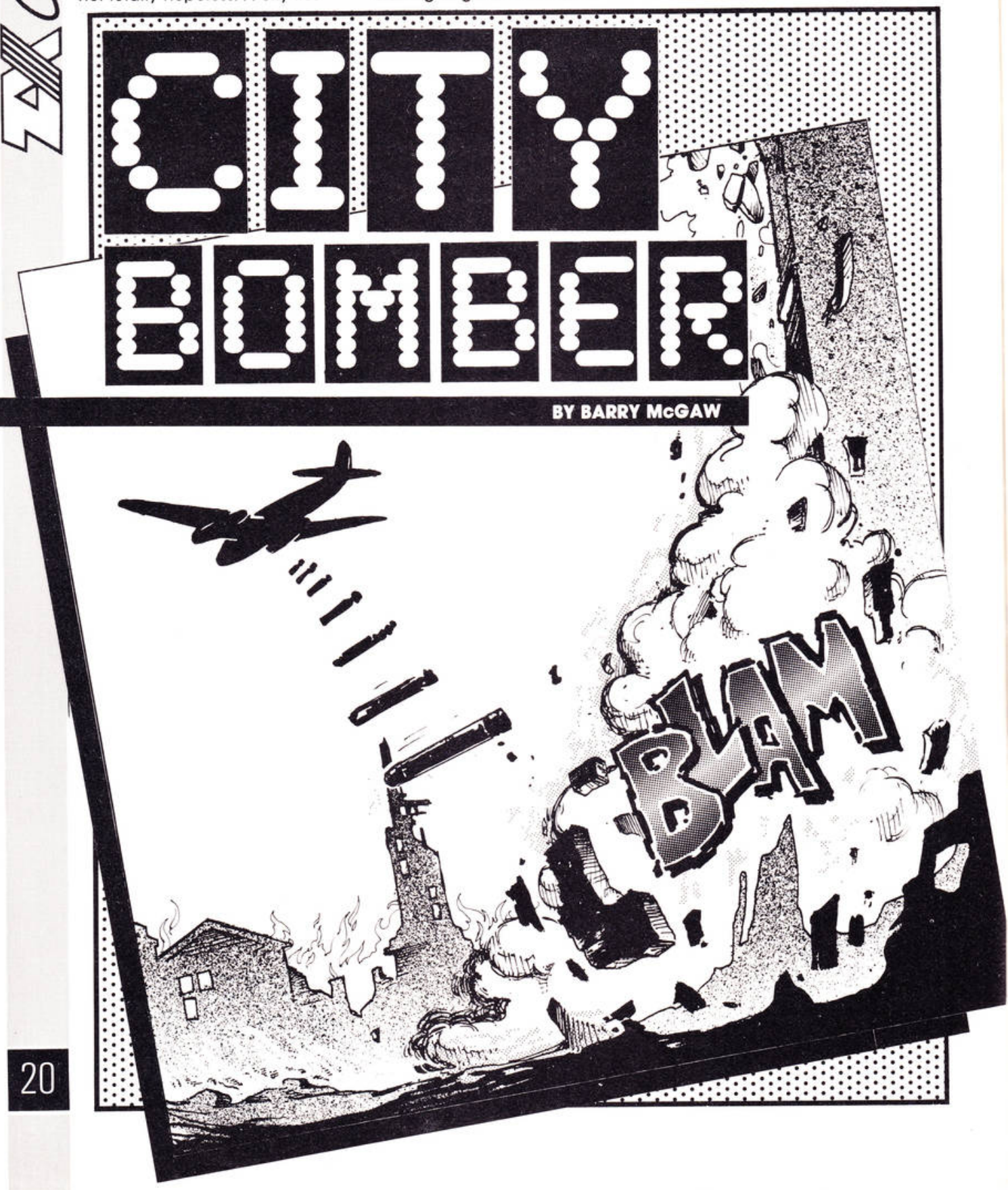
You must bomb the city in order to land your aircraft, but it's not quite that simple!

Imagine you are flying your aircraft and you have suddenly run out of fuel. DON'T PANIC! Even though there is not a runway for miles, the situation is not totally hopeless. A city lies below, and in order to land you must bomb this city. When you have successfully completed your task, the pilot gets out to stretch his legs, gets back into the plane and flies off only to find himself in the same situation again over another city!

Head in the clouds

However, this time things are starting to get a little more difficult: there are clouds to contend with. The clouds do not hurt you but stop your bombs until you are right above them. However, they do have advantages . . .

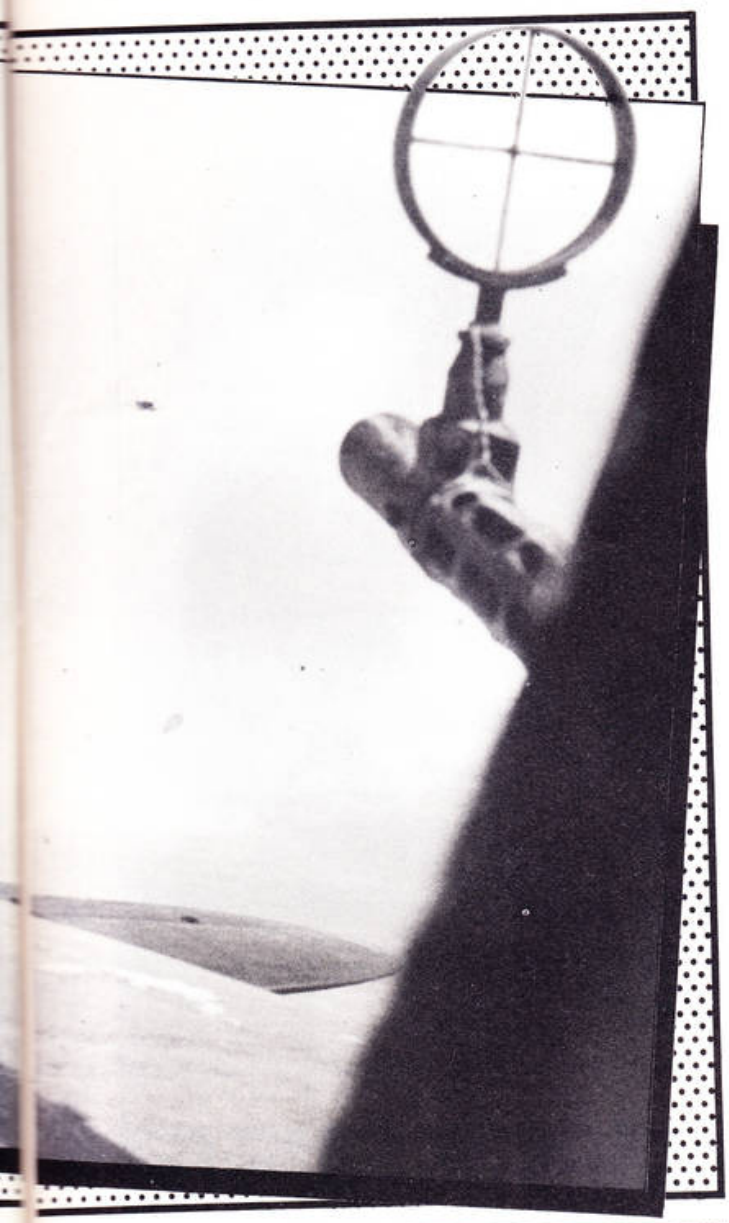
There are 19 skill levels, five speeds, a high-score table, 18 graphics characters and lots more in this great action-packed game. In order to drop your bombs, merely press Enter. Have fun!



```
  1 GO SUB 9600: LET a$="BAZ Mc
GAW": LET hs=0: BRIGHT 0: BORDER
 0: PAPER 5: INK 0: CLS
  2 LET l=3
  3 LET s=0
  5 POKE 23562,1
  10 LET a=1
  11 PRINT AT 10,11; INK 3; FLAS
H 1;"CITY BOMBER"
  12 PRINT AT 12,11;"BARRY MCGAW
"
  14 PRINT AT 20,0;"PRESS ANY KE
Y TO PLAY"
  16 IF INKEY$="" THEN  GO TO 15
  17 POKE 23609,6
  20 INPUT "Level(1 TO 19):";le
  21 IF le>19 OR le<1 THEN  GO T
O 20
  23 INPUT "ENTER SPEED(1 TO 5)"
;sp
  24 CLS : IF sp<1 THEN  IF sp>5
 THEN  GO TO 23
  25 PRINT AT 0,0;"PLEASE WAIT F
OR CITY GENERATION"
  30 FOR n=2 TO 28
  31 PRINT AT 21,n;"H"
  32 NEXT n
  40 LET r=INT (RND*4)
  50 LET n=27
  60 LET v$=STR$ le
  70 LET lle=LEN v$
 100 LET f$="E": LET g$="F"
 110 LET st=INT (RND*5): IF st=0
 OR st>3 THEN  GO TO 100
 120 LET i=INT (RND*21): IF i<le
+1 THEN  GO TO 120
 130 PRINT AT i,n; INK r;f$
 140 FOR f=21 TO i+1 STEP -1
 150 PRINT AT f,n; INK r;g$
 170 NEXT f
 175 LET r=INT (RND*4)
 180 LET n=n-st
 181 IF n<3 THEN  GO TO 201
 190 LET f$="C": LET g$="D"
 200 GO TO 110
 201 PRINT AT 0,0; INK 0;"SCORE"
;"000000";" LEVEL";"  ";" SPEED"
;sp;" MEN ";l
 202 PRINT AT 0,19-lle;le
 205 LET bd=a+1
 207 LET bm=0
 209 LET f12=0
 210 FOR b=28 TO 1 STEP -1
 212 LET s$=STR$ s
 215 LET ls=LEN s$
 216 PRINT AT 0,11-ls;s
 218 PRINT AT 0,31;l
 220 PRINT AT a,b; INK 0;"AB"
 230 PRINT AT a,b+2;"  "
 245 BEEP .006,-a: IF sp>1 THEN
 FOR n=1 TO (sp*2): NEXT n
 246 PRINT AT a,b+2;" "
 247 IF bm=5 THEN  GO TO 260
 250 IF CODE INKEY$=13 THEN  LET
 bm=bm+1: LET t=b: LET bd=a+1: G
O SUB 1000
 262 IF ATTR (a,b-1)=47 THEN  GO
 SUB 4000
 265 IF SCREEN$ (a,b-1)<>" " THE
N  GO SUB 2000
 267 IF l<1 THEN  PRINT AT 10,12
; INK 6; PAPER 2; FLASH 1;"Game
Over": GO TO 9000
 268 IF f12=0 THEN  GO TO 270
 269 RETURN
 270 NEXT b
 271 PRINT AT a,0;"    "
 280 LET a=a+1
 290 IF a=22 THEN  GO TO 310
 300 GO TO 210
 310 PRINT AT 21,0;"AB"
 320 PRINT AT 20,0;"G": FOR n=1
TO 5: BEEP .01,n: NEXT n: PAUSE
10: PRINT AT 20,0;" "
 330 FOR n=2 TO 30
 340 BEEP .006,n-12
 345 BEEP .004,n-5
 350 PRINT AT 21,n;" G"
 360 PAUSE 10
 400 NEXT n
 410 PRINT AT 21,31;" ": PRINT A
T 20,31;"G": BEEP .05,10: PAUSE
10: PRINT AT 20,31;" ": PRINT AT
 21,30;"▄": PAUSE 5
 420 FOR n=30 TO 2 STEP -1
 430 PRINT AT 21,n;"U "
 440 BEEP .004,n-5
 460 BEEP .006,n-12
 465 PAUSE 10
 500 NEXT n
 510 PRINT AT 21,2;" ": PAUSE 10
: PRINT AT 20,0;"U "
 515 BEEP .05,1
 520 PRINT AT 20,0;" "
 530 PRINT AT 21,0;"B ": PAUSE 6
: PRINT AT 21,31;"A": PAUSE 6
 540 PRINT AT 21,0;" ": PRINT AT
 21,30;"AB": PAUSE 6
 550 FOR j=20 TO 1 STEP -1
 560 PRINT AT 21,30;"  "
 570 PRINT AT j+1,j+1;"  ";AT j,
j;"AB"
 575 BEEP .006,-j
 576 NEXT j
 580 CLS
 590 PRINT AT 7,7; INK 7;"B";AT
```

```
8,6;"QPQ"
 600 PRINT AT 4,12; INK 7;"R";AT
 5,11;"QPQ"
 610 PRINT AT 5,20; INK 7;"R";AT
 6,19;"QPQ"
 620 PRINT AT 7,26; INK 7;"R";AT
 8,25;"QPQ"
 630 LET a=1: BRIGHT 0: GO TO 25
 999 GO TO 9000
1000 LET tp=bm
1001 LET f1=0
1002 LET err=0
1006 PRINT AT bd,t;"I": LET bd=b
d+1
1016 LET bm=5: LET f12=1
1017 GO SUB 270
1018 IF ATTR (bd,t)=47 THEN  GO
SUB 5000
1019 IF err=1 THEN  GO TO 1100
1020 IF SCREEN$ (bd,t)<>" " THEN
  LET s=s+2: GO SUB 6000: GO TO
1022
1021 IF bd=21 THEN  GO SUB 6000
1022 IF f1=1 THEN  GO TO 1100
1025 IF a=20 THEN  LET bd=20
1030 IF bd>21 THEN  GO TO 1100
1080 PRINT AT bd-1,t;" "
1090 GO TO 1006
1100 LET bd=a+1: LET f12=0: RETU
RN
2000 PRINT AT a,b;"KJ": FOR n=1
TO 4: BEEP .001,n: NEXT n: PRINT
 AT a,b;"LK": FOR n=1 TO 4: BEEP
 .002,n: NEXT n: PRINT AT a,b;"K
K": FOR n=1 TO 4: BEEP .003,n: N
EXT n
2020 PRINT AT a,b;"  "
2025 LET l=l-1
2030 RETURN
4000 BEEP .006,-a
4010 PRINT AT a,b;"B "
4011 BEEP .006,-a
4020 PRINT AT a,b;" "
4030 FOR n=b TO b-3 STEP -1
4040 BEEP .006,-a
4041 PAUSE 2
4050 NEXT n
4060 LET b=b-4
4070 PRINT AT a,b;"A": BEEP .006
,-a: PRINT AT a,b-1;"AB": BEEP .
006,-a: LET b=b-2: RETURN
5000 LET err=1
5010 PRINT AT bd-1,t;" "
5020 BEEP .07,-10
5030 LET bm=0
5100 RETURN
6000 LET bm=tp: PRINT AT bd-1,t;
" ": PRINT AT bd,t;"J": FOR n=1
TO 2: BEEP .006,n+1: NEXT n: PRI
NT AT bd,t;"K": FOR n=1 TO 2: BE
```

```
EP .006,-n: NEXT n: PRINT AT bd,
t;"N": FOR n=1 TO 2: BEEP .006,(
PI*-n): NEXT n: PRINT AT bd,t;"
": LET f1=1
6010 IF bm=5 THEN  LET bm=0
6020 RETURN
9000 FOR n=1 TO 40: NEXT n
9010 IF INKEY$="" THEN  GO TO 90
10
9030 PAPER 0: BORDER 0: CLS
9040 FOR n=0 TO 31 STEP 2
9050 PRINT AT 0,n; INK 3; PAPER
6; FLASH 1;"▬"
9060 PRINT AT 21,n; INK 6; PAPER
 3; FLASH 1;"▬"
9070 NEXT n
9080 FOR n=1 TO 30 STEP 2
9090 PRINT AT 0,n; INK 6; PAPER
3; FLASH 1;"▬"
9100 PRINT AT 21,n; INK 3; PAPER
 6; FLASH 1;"▬"
9110 NEXT n
```



```
9120 FOR f=0 TO 21 STEP 2
9130 PRINT AT f,31; INK 1; PAPER
 5; FLASH 1;"▮"
9140 PRINT AT f,0; INK 5; PAPER
1; FLASH 1;"▮"
9150 NEXT f
9160 FOR f=1 TO 21 STEP 2
9170 PRINT AT f,31; INK 5; PAPER
 1; FLASH 1;"▮"
9180 PRINT AT f,0; INK 1; PAPER
5; FLASH 1;"▮"
9190 NEXT f
9200 PRINT AT 3,11; INK 4;"HIGH
SCORE";AT 6,12; INK 6;"HELD  BY"
9205 LET x=11+(5-((LEN a$)/2))
9210 PRINT AT 8,x; INK 7;a$
9215 LET ls=LEN (STR$ hs)
9220 PRINT AT 12,13; INK 6; FLAS
H 1;"000000";AT 12,19-ls;hs
9310 IF s>hs THEN  LET hs=s: GO
SUB 9500
9400 PAUSE 0: BRIGHT 0: PAPER 5:
 INK 0: CLS : GO TO 2
9500 INPUT "NAME:"; LINE a$
9510 IF a$="" THEN  GO TO 9500
9511 IF LEN a$>10 THEN  GO TO 95
00
9515 PRINT AT 8,11;"          "
9520 LET x=11+(5-((LEN a$)/2))
9521 PRINT AT 8,x; INK 6;a$
9525 LET ls=LEN (STR$ hs)
9530 PRINT AT 12,19-ls; INK 6; F
LASH 1;hs
9540 RETURN
9600 PAPER 7: CLS : PRINT AT 10,
10; INK 2; PAPER 4; FLASH 1;"Ple
ase Wait"
9601 FOR i=97 TO 114
9605 IF i=109 THEN  LET i=110
9610 FOR n=0 TO 7
9620 READ x
9630 POKE USR CHR$ i+n,x
9640 NEXT n
9650 NEXT i
9700 DATA 24,12,6,15,17,63,0,0
9701 DATA 0,1,3,255,255,226,48,2
4
9702 DATA 31,53,213,191,213,181,
223,191
9703 DATA 223,181,213,191,213,18
1,223,191
9704 DATA 63,121,170,255,185,249
,191,255
9705 DATA 191,121,170,255,185,24
9,191,255
9706 DATA 24,24,28,48,80,24,100,
70
9707 DATA 31,63,121,249,249,249,
255,255
9708 DATA 0,24,60,60,24,0,0,0
9709 DATA 0,0,32,74,138,71,153,7
4
9710 DATA 32,70,32,3,5,73,170,14
5: DATA 0,12,18,18,12,64,165,66
9711 DATA 0,8,1,32,0,64,8,0
9712 DATA 224,254,255,252,240,25
4,255,248
9713 DATA 255,255,255,255,255,25
5,255,62
9714 DATA 0,63,127,31,15,127,255
,31
9715 DATA 0,0,0,0,16,124,126,255
9716 FOR n=0 TO 7
9718 READ ax
9720 POKE USR "u"+n,ax
9730 NEXT n
9740 DATA 24,24,56,12,10,24,38,9
8
9750 RETURN
9760 REM **********************
9770 REM * Barry Mcgaw I.O.M *
9780 REM **********************
```

SOUND ANALYSER

BY DARREN ADKINSON

Analyse sound on your Spectrum. The potentials of this program include password recognition, start and stop signals and its use as a burglar alarm.

This program can be used to analyse a passage of music or speech which is recorded onto tape and then played into the computer. The computer scans the ear socket and every time a sound is heard, the program records a '1' in the string, otherwise a '0' is recorded. When the scanning is finished (after 1000 scans this period can be changed), the user can see a binary representation of the sound (a string holding 1s or 0s) or a symbolic representation (a trough which moves along the screen bearing a 'hump' where there is a sound).

Cutting a record

The scan is recorded in a string so that the recorded sound may be analysed. For example, the user could record himself saying the password to a program and then analyse the sound later.

Obviously the program has limited uses, but the important thing is that the Spectrum IN address 49150 (and others — see the manual) will know when a sound is coming through the ear socket. For this reason, the computer can, in a way, be controlled by sound. For instance, you could use a microphone connected to the ear socket to start or stop a program. People may like to experiment with more control. When using a microphone, an amplifier should be used to boost the signal (make sure that this is not too great, otherwise it may damage the computer).

Stop, thief!

Also, the Spectrum can be used as a burglar alarm. If a microphone is placed near a door then the noise from the door opening could trigger the Spectrum to produce a warning signal. This is only a simplified system but we are sure that there are many readers who, with a bit of ingenuity, can produce a good alarm system.

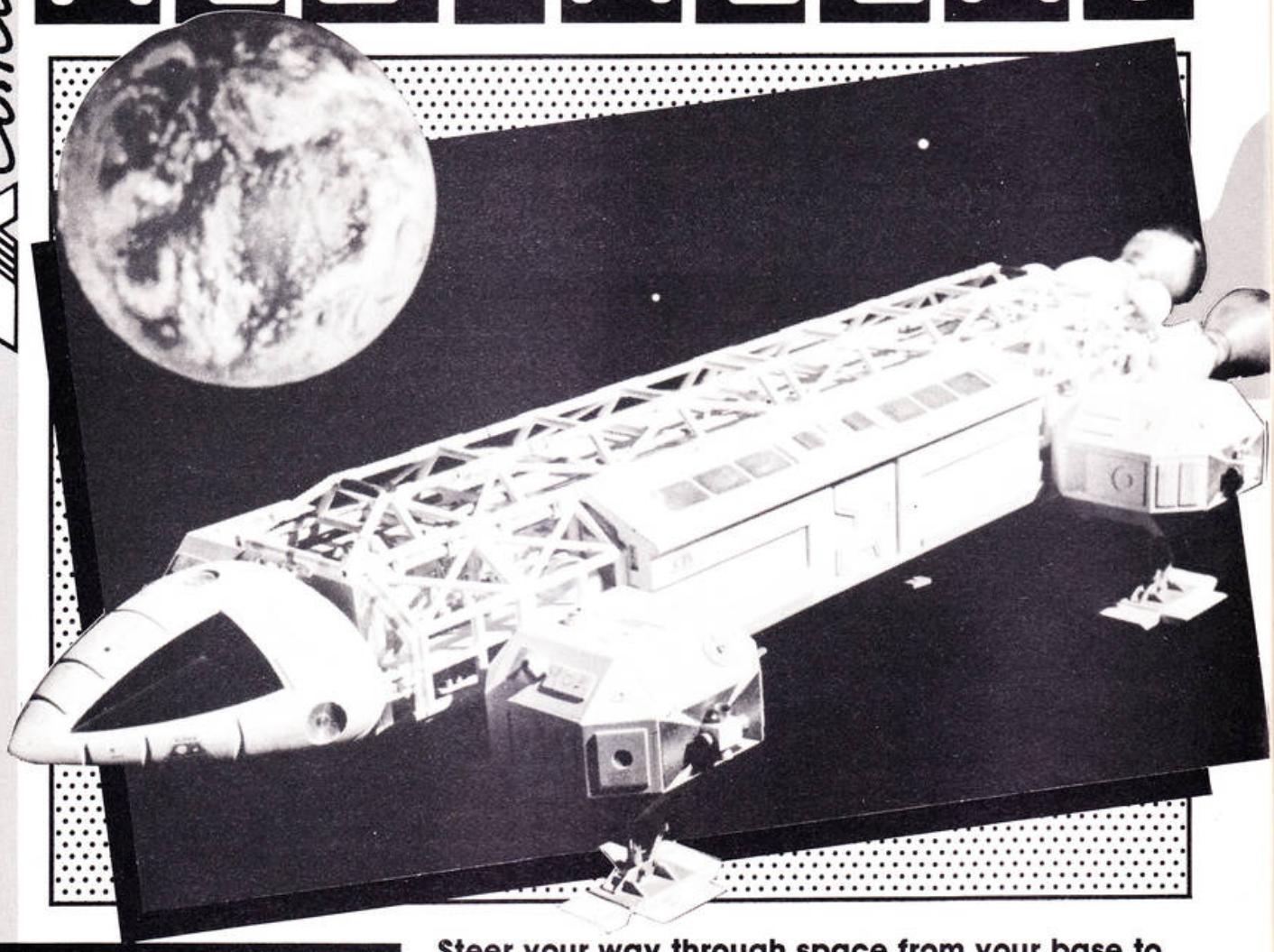
```
  5 POKE 23658,8
 10 BORDER 0: PAPER 0: INK 7
 20 CLS
 30 PRINT  INK 6; PAPER 1;"SOUN
D ANALYSING PROGRAM."
 40 PRINT : PRINT
 50 PRINT  INK 6; PAPER 1;"TO B
EGIN PRESS PLAY ON TAPE      RECO
RDER AND KEY ""1"""
 60 IF INKEY$<>"1" THEN  GO TO
60
 70 CLS
 80 PRINT  INK 6; PAPER 1; FLAS
H 1;AT 0,5;"SOUND SCAN IN PROGRE
SS."
 90 DIM A$(1000)
100 FOR B=1 TO 1000
110 LET C=IN 49150
120 IF C=191 THEN  LET A$(B)="0
"
130 IF C<>191 THEN  LET A$(B)="
1"
140  NEXT B
150 PRINT  INK 6; PAPER 1;AT 0,
0;"SYMBOLIC OR BINARY DISPLAY?(S
/B)"
160 INPUT D$
170 IF D$="S" THEN  GO TO 200
180 IF D$="B" THEN  GO TO 350
190 GO TO 160
200 LET E=11
210 LET F=0
215 CLS
220 FOR G=1 TO 1000
230 IF A$(G)="0" THEN  PRINT AT
E,F;"_": PRINT AT E,F+1;" "
240 IF A$(G)="1" THEN  PRINT AT
E,F;"▒": PRINT AT E,F+1;" "
250 LET F=F+1
260 IF F=31 THEN  LET F=0
270 NEXT G
280 PRINT  INK 6; PAPER 1;AT 21
,0;"SEE DISPLAY AGAIN OR RUN (D/
R)"
290 INPUT H$
300 IF H$="D" THEN  CLS : GO TO
150
310 IF H$="R" THEN  RUN
320 GO TO 290
350 CLS
360 PRINT A$
370 PRINT : PRINT
380 PRINT  INK 6; PAPER 1;"HARD
COPY? (Y/N)"
390 INPUT I$
400 IF I$="Y" THEN  LPRINT A$:
GO TO 280
410 IF I$="N" THEN  GO TO 280
```

```
420 GO TO 390
500 FOR J=0 TO 7
510 READ K: POKE USR "L"+J,K
520 NEXT J
530 DATA BIN 00000000
540 DATA BIN 01111110
550 DATA BIN 01000010
560 DATA BIN 01000010
570 DATA BIN 01000010
580 DATA BIN 01000010
590 DATA BIN 01000010
600 DATA BIN 11000011
```

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000001000000000000000
00000010000000000000000000000000
01000000000011111000001111111111
01111100000000011111000000000000
01100111111010000011111110000000
01111111111111000000000000000000
00001111111111111101110000000000
00000000000000000001001010000000
00000000000000000000000000001111
11000001111111000000000000010000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000010001100000000001
01100010000000111111111110000000
00000000000001101010111110100010
10000101111110110110000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000
```

Binary display of the Consultant Editor singing.

RED ALERT



BY M S HARRIS

Steer your way through space from your base to the target.

Red Alert was written on a 48K Spectrum, but should work well on a 16K machine. A machine code scrolling routine and the UDG data must be entered, but read on and all will be revealed!

Up and running

When the game is up and running, you should see the title page; wait until you are asked to press 'S'. You should now see your ship with some stars scrolling towards it. You have to dodge these stars, but you can shoot them to clear your way.

There are two phases to the game, the latter being the harder of the two. In phase one your laser will destroy a whole line of debris, whereas in phase two it will only destroy a single piece. In phase one you have 20 shots, in phase two you have only 10 and you also have to travel twice as far. You have three ships with which to complete your mission; you can lose the game by either running out of ammunition or by hitting a piece of debris.

So, full 'steam' ahead, Captain — this is a full-scale RED ALERT.

But, first...

The first thing to do is to enter the UDG data into memory — use the program in Figure 1. Enter the data carefully. Now you must save these characters and you do this by typing:

SAVE "GRAPHICS" CODE USR "A",128

Now verify them by using VERIFY "" CODE

O.K? Type NEW, as the graphics are above RAMTOP.

Now the machine code scrolling routine must be entered. For this use Figure 2; again type in the data carefully and check it over. Now run this program and save the code by using:

SAVE "SCROLL" CODE 30000,25

Now verify using VERIFY "" CODE

All O.K? THEN TYPE randomise user 0 and reload the 2 pieces of code (remember to clear 29999). Now finally, the main BASIC program must be typed in, checked and saved by using

SAVE "RED ALERT" LINE 5

Now verify it using VERIFY"".

All O.K? Then type RUN and you're off!

LINE BY LINE

10-140	Loop for the title page.
150	Clear screen 2 spaces at a time.
160-18-0	Final message routine.
200	Set up main variables.
230	Checks if your ship has crashed. If so takes 1 ship away and if there are no ships left, goes to the appropriate page.
250-260	Set up barriers.
280-310	Checks keyboard for movement, FIRE, etc.
320-330	Stops your ship going out of the barriers.
340	Increases your score by 1 and scrolls the screen.
350-360	Stops the debris going out of the barriers.
370	Prints the debris.
390-400	Checks the length of the universe and, when appropriate, goes on to phase two.
410	End of main loop.
420-510	Decrease ammunition by 1 and fire shell. Checks if there is any ammunition left.
530-550	End of phase 1, entering phase 2 routine.
560	Set up new values for variables.
590	As 230
600	Set up barriers.
630-660	As 280-310.
670-680	As 320-330.
700-710	Checks the debris doesn't go out of the barriers.
750	End of main loop.
770-860	As 420-510.
890	Page when all 3 ships have been destroyed.
980	Page when all ammunition has been used.
1080	End of game routine (the out 254,F produces the border effect).

CONTROL KEYS

1	1 TO MOVE UP.
Q	TO MOVE DOWN.
0	TO FIRE THE LASER.
H	HOLDS THE GAME, UNTIL THE NEXT KEY IS PRESSED.

To make loading easier a loader could be used. It could be something like this:

```
10  CLEAR 29999.
20  LOAD ""CODE.
30  LOAD ""CODE.
40  LOAD "".
```

```
  20 BORDER 0: PAPER 0: INK 7: C
LS
  30 LET A$="
  40 PRINT AT 7,0;"
                   "
  50 PRINT AT 8,0;"
                   "
  60 PRINT AT 1,1; INK 2; BRIGHT
 1;a$
  70 BRIGHT 1: INK 6: LET y=12:
LET x=15: LET b$="By

          Michael Harris 1984


       PRESS "S" TO START THE GAM
```

RED ALERT

```
  1 REM      Fig 1
  2
 10 FOR f=USR "a" TO USR "c"+7
 20 READ a: POKE f,a: NEXT f
 30 DATA 112,124,255,255,255,25
5,124,112,0,0,224,255,255,224,0,
0,0,224,28,127,127,28,224,0
```

```
  1 REM      Fig 2
  2
 10 CLEAR 29999: LET A=30000
 11 LET A$="06C011004ED5E123C50
11F001AED6026770023231 3C110F0C9"
 20 FOR F=A TO 30024
 30 POKE F,16*(CODE A$(1)-48-(7
*(A$(1)>"9")))+CODE A$(2)-48-(7*
(A$(2)>"9"))
 40 LET A$=A$(3 TO ): NEXT F
```

```
 80 FOR i=1 TO LEN b$: PRINT AT
 y,x;b$( TO i);: BEEP .05,1: NEX
T i
 90 LET c=1
100 IF INKEY$="s" OR INKEY$="S"
THEN  GO TO 150
110 BRIGHT 1: INK c: PRINT AT 1
,1;a$
120 PAUSE 20: LET c=c+1
130 IF c>=8 THEN  LET c=1
140 GO TO 100
150 PRINT AT 0,0;: FOR q=0 TO 1
91: PRINT "    ";: BEEP .01,-2: N
EXT q
160 PRINT AT 5,11; INK 1; PAPER
6; FLASH 1;"GOOD-LUCK"
170 PRINT AT 10,0; PAPER 2; INK
9;"   (you will sure need it!)
  "
180 FOR S=0 TO 50: BEEP .1,S: N
EXT S
190 BRIGHT 1: BORDER 0: PAPER 0
: INK 7: CLS
200 LET count=0: LET li=3: LET
am=20: LET sc=0: LET up=0: LET d
own=8
210 LET c=RND*15
220 PRINT AT down,up; INK 4;"BB
"
230 IF SCREEN$ (down,up+2)="*"
THEN  LET li=li-1: FOR g=0 TO 5:
 FOR f=0 TO 60 STEP 10: BEEP .01
,f: NEXT f: NEXT g: IF li<=0 THE
N  GO TO 880: GO TO 190
240 PRINT AT down,up; INK 4;"BB
"
250 PRINT AT 0,0; INK 3; BRIGHT
1;"_____
```

```
_____"
260 PRINT AT 16,0; INK 3; BRIGH
T 1;"_____
_____"
270 PRINT AT ABS down,ABS up;"
 "
280 IF INKEY$="1" THEN  LET dow
n=down-1
290 IF INKEY$="h" OR INKEY$="H"
 THEN  PAUSE 0
300 IF INKEY$="q" OR INKEY$="Q"
 THEN  LET down=down+1
310 IF INKEY$="0" THEN  GO SUB
420
320 IF down>=15 THEN  LET down=
15
330 IF down<=1 THEN  LET down=1
340 LET SC=SC+1: RANDOMIZE USR
30000
350 IF c>=16 THEN  LET c=1
360 IF c=0 THEN  LET c=1
370 RANDOMIZE : PRINT AT c,20;
INK 9; BRIGHT 1; INK 7;" *  *  *
 "
380 PRINT AT down,31;" "
390 LET count=count+1
400 IF count>=400 THEN  GO TO 5
20
410 GO TO 210
420 LET am=am-1: FOR f=1 TO 20
430 PRINT AT down,f; INK 5;"C"
440 PRINT AT down,30;" "
450 PRINT AT down,up; INK 4;"BB
"
460 PRINT AT down,f;" "
470 IF am=0 THEN  PRINT AT 5,7;
"OUT OF AMMUNITION": FOR f=0 TO
500: NEXT f: GO TO 970
480 PRINT AT down,up; INK 4;"BB
"
490 NEXT f
500 PRINT AT down,30;" "
510 BORDER 0: RETURN

520 REM ~SCREEN 2~
530 BORDER 0: PAPER 0: CLS : BR
IGHT 1: PRINT AT 10,0; FLASH 1;
PAPER 6; INK 2;"          PHASE 1
COMPLETE          "
540 FOR f=0 TO 20: BEEP .1,4: B
EEP .1,2: BEEP .01,5: BEEP .1,1:
 NEXT f
550 PRINT AT 10,0; FLASH 1; BRI
GHT 1; PAPER 1; INK 5;"          E
NTERING PHASE 2          ": FOR F=
0 TO 400: NEXT F: CLS
560 LET am=10: LET count=0
570 LET c=RND*15
```

```
 580 PRINT AT down,up; INK 4; BR
IGHT 1;"BB"
 590 IF SCREEN$ (down,up+2)="#"
THEN  LET li=li-1: FOR g=0 TO 5:
 FOR f=0 TO 60 STEP 10: BEEP .01
,f: NEXT f: NEXT g: IF li<=0 THE
N  GO TO 880: GO TO 190
 600 PRINT AT 0,0; INK 3; BRIGHT
 1;"_____
____";AT 16,0;"_____
_____"
 610 PRINT AT down,up; INK 4; BR
IGHT 1;"BB"
 620 PRINT AT ABS down,ABS up;"
 "
 630 IF INKEY$="1" THEN  LET dow
n=down-1
 640 IF INKEY$="h" OR INKEY$="H"
 THEN  PAUSE 0
 650 IF INKEY$="q" OR INKEY$="Q"
 THEN  LET down=down+1
 660 IF INKEY$="0" THEN  GO SUB
770
 670 IF down>=15 THEN  LET down=
15
 680 IF down<=1 THEN  LET down=1
 690 INK 9: LET SC=SC+1: RANDOMI
ZE USR 30000
 700 IF c>=15 THEN  LET c=1
 710 IF c=0 THEN  LET c=1
 720 RANDOMIZE : PRINT AT c,10;
INK 9;" # "
 730 LET count=count+1
 740 PRINT AT down,31;" "
 750 IF count>=700 THEN  GO TO 1
000
 760 GO TO 570
 770 LET am=am-1: FOR f=1 TO 10
 780 PRINT AT down,f; INK 5;"S"
 790 IF SCREEN$ (down,f+2)="#" T
HEN  PRINT AT down,f;"  ": FOR
f=10 TO 20: BEEP .01,f: NEXT f
 800 PRINT ;AT down,up; INK 4;"B
B"
 810 PRINT ;AT down,up; INK 4;"B
B"
 820 PRINT AT down,f;"  "
 830 IF am=0 THEN  PRINT AT 5,7;
"OUT OF AMMUNITION": FOR f=0 TO
300: NEXT f: GO TO 970
 840 PRINT ;AT down,up; INK 4;"B
B"
 850 NEXT f
 860 BORDER 0: RETURN
 870 STOP
 880 REM CRASH PAGE
 890 BRIGHT 0: BORDER 1: PAPER 1
: INK 7: CLS
 900 PRINT AT 2,1;"YOU HAVE DEST
ROYED ALL 3 SHIPS."
 910 PRINT AT 4,1;"THE ALIENS HA
VE TAKEN  OVER THE  PLANET."
 920 PRINT AT 8,1;"YOUR FINAL SC
ORE WAS ";sc;" pts"
 930 PRINT AT 15,1;"WOULD YOU LI
KE TO TRY AGAIN ?"
 940 PRINT AT 17,1;"IF YOU HAVE
YHE NERVE PRESS ~Y~"
 950 IF INKEY$="y" OR INKEY$="Y"
 THEN  GO TO 190
 960 GO TO 950
 970 REM OUT OF AMMO PAGE
 980 BRIGHT 0: BORDER 1: PAPER 1
: INK 7: CLS
 990 PRINT AT 2,0;"YOU BUMBLING
FOOL !
                    YOU USED ALL
OF YOUR PHOTON      LASERS!."
1000 PRINT AT 8,0;"WHAT A STATE
TO GET YOURSELF IN!"
1010 PRINT AT 10,0;"THE ALIENS H
AVE TAKEN OVER THE  PLANET."
1020 PRINT AT 13,0;"DO YOU WANT
TO TRY ONCE MORE ?"
1030 PRINT AT 15,0;"IF SO BE MOR
E CAREFULL,PLEASE!"
1040 PRINT AT 18,0;"YOU SOME HOW
 SCROUNDED ";sc;" pts"
1050 PRINT AT 20,4;"PRESS ~Y~ TO
 TRY AGAIN"
1060 IF INKEY$="y" OR INKEY$="Y"
 THEN  GO TO 190
1070 GO TO 1060
1080 BRIGHT 0: BORDER 0: PAPER 0
: INK 9: CLS
1090 PRINT AT 1,8; INK 9; FLASH
1;"CONGRATULATIONS"
1100 FOR g=0 TO 1: FOR f=0 TO 25
5
1110 OUT 254,f
1120 NEXT f: NEXT g
1130 BORDER 0: INK 9: PRINT AT 3
,0;"WELL DONE:- YOU HAVE SAVED O
UR  PLANET FROM CERTAIN DEATH."
1140 PRINT AT 8,0;"YOU ARE THE W
ORLDS GREATEST HERO"
1150 PRINT AT 8,0; OVER 1;"_____
_____"
1160 PRINT AT 12,5;"CAN YOU DO I
T AGAIN ?";AT 14,6;"PRESS ~S~ TO
 START"
1170 PRINT AT 21,2;"YOUR CHAMP S
CRORE WAS ";sc;" Pts."
1180 IF INKEY$="S" OR INKEY$="s"
 THEN  GO TO 190
1190 GO TO 1180
```

Read on and a fantastic screen format/data base utility will unfold before your very eyes!

This program is a screen format/data base utility for on screen data capture and information storage, sorting, editing, enquiry and general manipulation. The screen format section of the program is designed to emulate what is commonly available on larger business micros.

Operating Instructions

The program on loading presents the user with 10 options available for use. The program is idiot-proof in as much as it will not allow you to use an option if it is not logical to use the option at the stage — eg: trying to display or sort data which does not exist. A logical sequence of using the program could be as follows:

1. OPTION 1: Instructions and Scope
This option will list a summary of what the the SF-UTIL program does as an overview to the user. It is not intended to be a full set of instructions. It also shows an example of a formatted screen

2. OPTION 2: Format Screen
This option will display a sub menu which will allow you to format your screen with fixed text and variable data definition and then return to the main menu.

OPTION 1: Text on a blank screen.
Use cursor keys to move the cursor. Use edit to move the cursor to start of next line. Use caps/lock to change from upper to lower case and vice versa. Press ENTER to accept your screen. This is where you enter the fixed text portion of your screen — eg: those portions that remain the same each time you capture information (headings, titles, underlinings, etc).

OPTION 2: Variable Data Fields.

The idea here is to place the cursor at the position of the first field to be captured and then press ENTER. The program will then ask you to input field length (from 1 to 30), paper (0-7), ink (0-7), bright (0 or 1), flash (0 or 1) and inverse (0 or 1). Thereafter it prints a row of X's the length of the field that you specified with the colour attributes which you entered. If this looks correct you press ENTER to continue to the next field to be defined or you press 0 (Zero) to wipe out the current field if it is wrong to allow you to try again or move the cursor to a more desirable position and try again.

When you have moved your cursor to all the fields that need defining you press DELETE to return you to the sub-menu.

Remember one important factor. If you re-enter the Option 2 phase of this sub-menu and try to define new variable data fields, all data previously captured in the machine will be cleared out of memory. If you, therefore, wish to save such data you must first use the SAVE DATA option of the main menu. Obviously if no data has as yet been captured then you lose nothing. Up to 40 variable fields are permitted.

OPTION 3: Display Finished Screen

This option will display the completed screen which you will be presented with when you enter the data capture phase of the system. If any aspects of this screen are not to your liking you can re-enter Options 1 and/or 2 of the sub menu to re-define the **entire** screen of text and/or variable data.

OPTION 4: Display Data Definition

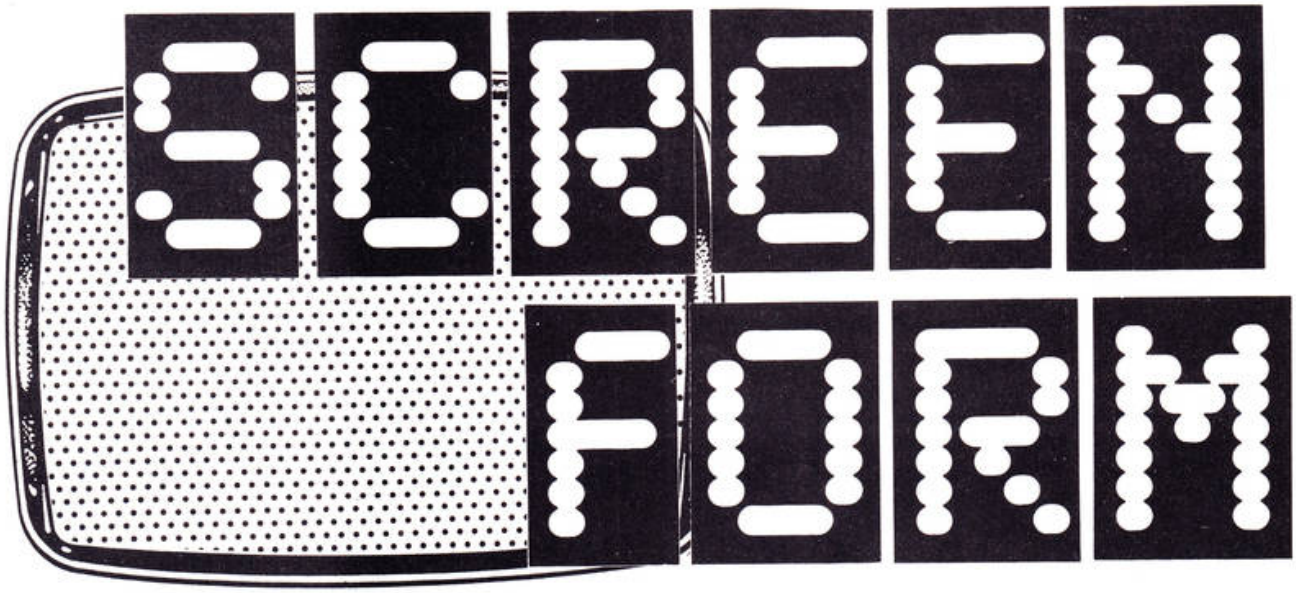
This option will display a full definition of the data. This definition is held in memory in an assay T (40,9). Each variable field is defined here. Up to 40 variable fields are permitted and each has 9 parameters viz:

1. Row position on screen.
2. Column position on screen.
3. Length of field in characters.
4. Start position of data in the array D$

5-9 Paper Ink Bright Flash Inverse

This definition will be extremely useful to the budding

BY ALISTAIR DE WET



programmer, as it allows access to any of the stored data (in D$ array) at field level for extra program codes to be included in this program or in other programs which he may write to read the data which has been saved from this program. If you are going to write your own program, I would suggest that you merge the load procedure from ST-UTIL into your program which will save you time and trouble, as there are 3 files to load into memory (FIXED TEXT ARRAY, TABLE ARRAY, DATA ARRAY). Also the data array size is governed by the variable fields as defined in the table array and is thus set up in the load subroutine at run time.

OPTION 5: Return to Main Menu
Does just that.

3. OPTION 3: Formatted Data Capture
This allows you to capture data according to the definition you set up in the Option 2 sub menu. The cursor will position itself on the first column of the first field to be captured. Start entering information. The cursor moves automatically from one field to the next as it fills up the field. To move the cursor from one field to the next without entering data use the ↑ and ↓ cursor keys for next and prior fields.

To move the cursor from one character to another use the and cursor keys for next and prior character. Caps lock will change from upper to lower case and vice versa.

Press ENTER once you have entered all the data you want to accept the screen data and store it in memory.

4 OPTION 4: Edit/Enquiry
The first thing you are asked to do is to enter the number of the field on which you wish to enquire. This action will cause the appropriate field to flash. You are then asked if this is correct and you have a chance to change your mind or correct your choice. You will then be asked to input the string of data you wish to enquire on. This could be the entire field length or just part of a field — ie, you may wish to enquire on the surname field and look for all surnames starting with "SM". You would just enter "SM". The enquiry system will search the entire surname field looking for "SM". As soon as the first record is found containing an "SM" within the surname the document will be displayed with options at the bottom of the screen that read 1 ALT 2 DEL 3 PRT 4 CONT 5 QUIT.

If you enter:

(1) You can alter the contents of the record using the same method of moving the cursor and pressing ENTER after you have completed your changes as per data capture instructions.
(2) Will delete the current record which is displayed on the screen.
(3) Will copy the current record to the printer.
(4) Will search for further records with the same seek key (in this case "SM").
(5) Will not search any further but will return to main menu.

5. OPTION 5: Display Records
This option starts at the first record and displays all the records stored so far. Pressing any key will stop the display at the current record at which point the option is given to continue or return to main menu.

6. OPTION 6: Save Data
SAVE DATA saves the three arrays used by the system. You will have to press a key when the machine requests you to do so. The 3 arrays are interdependent upon each other in the controlling of the data handling of the system.

7. OPTION 7: Load Data
Enter the name of the file to be loaded or just press ENTER to load any file.

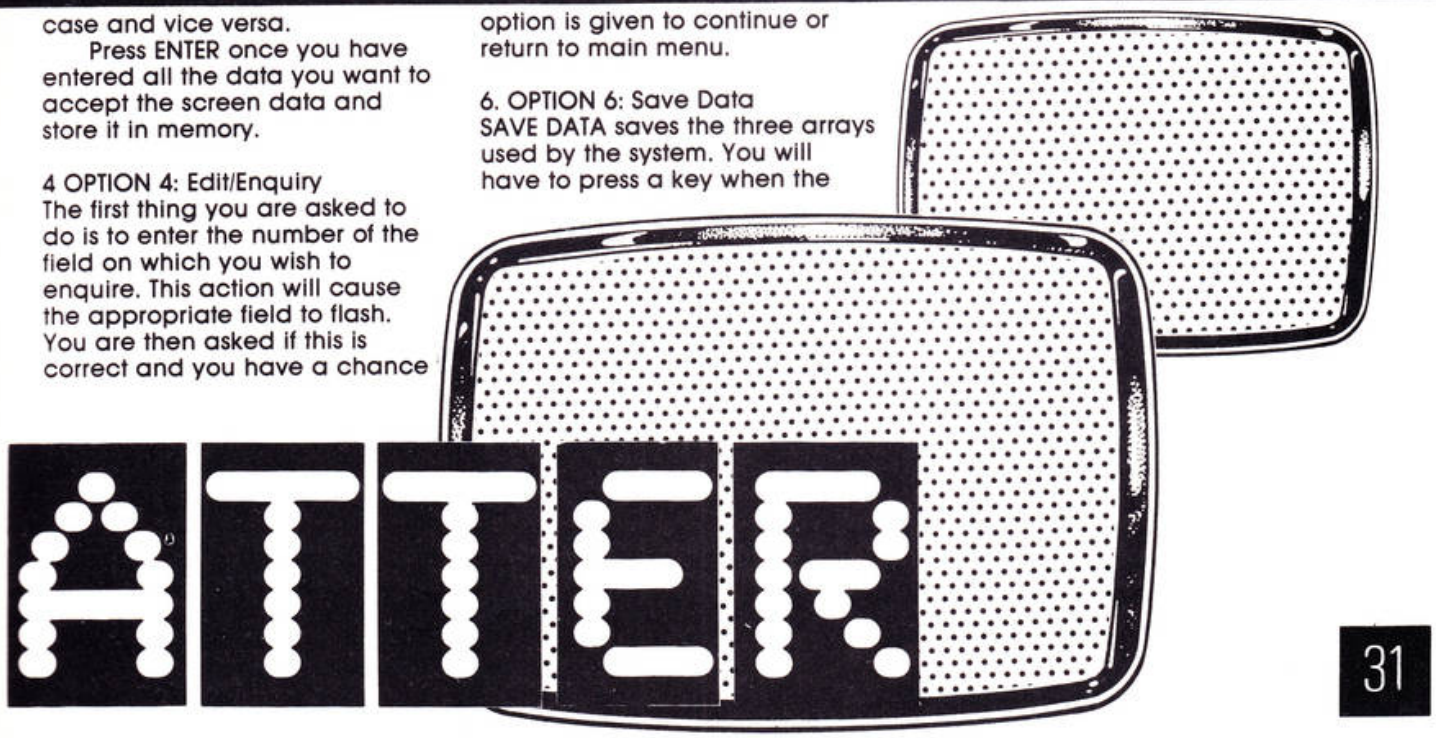
3 arrays previously saved will not be loaded. This load module will obviously overwrite any formats or data presently in the machine.

9. OPTION 9: Sort the Data
The first thing you are asked to do is to input the number of the field on which you wish to sort. This section will cause the appropriate field to flash. You are then asked if this is correct and you have the chance to correct your choice. Once you are satisfied the sort begins. Entering Option 5 (Display Data) will show that data has been correctly sorted.

10. OPTION 10: End Routines
This routine will give the user the option to stop the run or to go back to the main menu if he has forgotten to save his data.

NOTE

The program is, under normal operating conditions, not likely to crash. If, however, you manage to crash the program Type **GO TO 500 Do not type RUN or all your data will be lost.**



VARIABLES USED

DIM S$ (1,DIMY)	— For sorting
DIM F$ (1,704)	— Text
DIM T (100,9)	— Contains Data Definition
DIM D$ (DIMX,DIMY)	— Contains Captured Data
A	— Main Menu Option
B	— Sub Menu Option
X+Y	— Cursor Position
NT	— No. in DIMT
ND	— No. of documents in DS
PT	— Position in T array
C thru J	— General used
A$ & B$	— General used
MW	—
M$ Array	— Used in Data Accumulation Routine
NM	—

```
 10 REM ************************
 12 REM CONTROL MODULE
 14 REM ************************
 20 GO SUB 200
 30 GO SUB 500
 90 CLS
100 STOP
200 REM ************************
202 REM INITIALISE ROUTINE
204 REM ************************
210 BORDER 0: PAPER 0: INK 7
220 CLS
225 LET DIMX=1: LET DIMY=1
230 DIM F$(1,704)
240 DIM T(40,9)
250 DIM D$(DIMX,DIMY)
260 LET J=0
270 LET NT=1
280 LET ND=1
290 POKE 23658,8: POKE 23609,50
300 LET SW=0: LET SW1=0
310 LET X$="XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX"
390 RETURN
502 REM MAIN PROCESSING
504 REM ************************
510 GO SUB 1000
600 IF A=1 THEN  GO SUB 2000
650 IF A=2 THEN  GO SUB 3000
700 IF A=3 THEN  GO SUB 5000
750 IF A=4 THEN  GO SUB 6000
800 IF A=5 THEN  GO SUB 7000
850 IF A=6 THEN  GO SUB 4000
870 IF A=7 THEN  GO SUB 4200
900 IF A=8 THEN  GO SUB 8000
950 IF A=9 THEN  GO SUB 9200
960 IF A=0 THEN  GO SUB 9000
980 GO TO 500
1000 REM ************************
1002 REM MAIN MENU
1004 REM ************************
1010 BORDER 0: PAPER 0: INK 7
1020 CLS
1050 PRINT AT 0,5;"SCREEN FORMAT
 UTILITY";AT 1,5;"~~~~~~~~~~~~~~
~~~~~~~~"
1060 PRINT AT 2,7; A F DE WET
1984";AT 3,7;"~~~~~~~~~~~~~~~~~~~~"
1070 PRINT AT 4,7;"MAIN MENU SCR
EEN";AT 5,7;"~~~~~~~~~~~~~~~~~~~~"
1080 PLOT 40,175: DRAW -40,0:
 DRAW 0,-175: DRAW 255,0:
  DRAW 0,175:  DRAW -48,0
1090 PRINT AT 7,2;"1  -  INSTRUC
TIONS & SCOPE";AT 8,2;"2  -  FOR
MAT SCREEN";AT 9,2;"3  -  FORMAT
TED DATA CAPTURE";AT 10,2;"4  -
 ENQUIRY/EDIT";AT 11,2;"5  -  DI
SPLAY DOCUMENTS";AT 12,2;"6  -
SAVE DATA TO   TAPE";AT 13,2;"7
    LOAD DATA FROM TAPE";AT 14,2
;"8  -  ACCUMULATE TOTALS";AT 15
,2;"9  -  SORT THE DATA";AT 16,2
;"0  -  END ROUTINE"
1100 PRINT AT 20,2; FLASH 1;"ENT
ER OPTION REQUIRED"
1110 LET A$=INKEY$
1115 IF A$="" THEN  GO TO 1110
1120 LET A=CODE A$
1130 IF A<48 OR A>57 THEN  GO TO
       1110
1140 LET A=A-48
1490 RETURN
2000 REM ************************
2002 REM INSTRUCTIONS
2004 REM ************************
2010 GO SUB 4800
2020 PRINT AT 3,2;"PLEASE READ Y
OUR MANUAL";AT 4,2;"WELL AS INST
UCTIONS STORED";AT 5,2;"AS HARD
CODE IN A PROGRAM";AT 6,2;"OF TH
IS NATURE ONLY WASTE ";AT 7,2;"S
PACE"
2030 PRINT AT 14,2; FLASH 1;"PRE
SS ANY KEY TO CONTINUE"
2040 PAUSE 0
2210 BORDER 7: PAPER 7: INK 0: C
LS
2220 PRINT AT 0,0; FLASH 1;" EXA
MPLE OF SCREEN FORMAT "
2230 PRINT AT 3,0;"ACCOUNT NO [X
XXXXX]"
2240 PRINT AT 5,0;"NAME        [X
XXXXXXXXXXXXXXXXXX]"
2250 PRINT AT 7,0;"BIRTH DATE [X
X/XX/XX]"
2260 PRINT AT 9,0;"SEX         [X
]  STATUS    [X]"
2270 PRINT AT 11,0;"JOIN DATE  [
XX/XX/XX]"
2280 PRINT AT 13,0;"OCCUPATION [
XXXXXXXXXXXX]"
2290 PRINT AT 15,0; INVERSE 1;"X
's REPRESENT VARIABLES"
2350 PRINT AT 20,2; FLASH 1;"PRE
```

```
SS ANY KEY TO CONTINUE"
2360 PAUSE 0
2400 RETURN
3000 REM **********************
3002 REM FORMAT SCREEN
3004 REM **********************
3010 GO SUB 4800
3020 PRINT AT 3,2;"1 - TEXT ON B
LANK SCREEN";AT 5,2;"2 - VARIABL
E DATA FIELDS";AT 7,2;"3 - DISPL
AY FINISHED SCREEN";AT 9,2;"4 -
DISPLAY DATA DEFINITION";AT 11,2
;"5 - RETURN TO MAIN MENU";AT 13
,10;"***BEWARE***";AT 14,2;"ENTE
RING OPTION 2 CLEARS";AT 15,2;"M
EMORY OF ALL DATA PREVIOUSLY";AT
 16,2;"CAPTURED. BE CAREFULL!!!
 "
3030 PRINT AT 18,2; FLASH 1;"SEL
ECT OPTION REQUIRED"
3040 LET A$=INKEY$
3050 IF A$="" THEN  GO TO 3040
3060 LET B=CODE A$
3070 IF B<49 OR B>53 THEN  GO TO
 3040
3080 LET B=B-48
3085 IF B=1 THEN  GO SUB 3100
3090 IF B=2 THEN  GO SUB 3300
3095 IF B=3 THEN  GO SUB 3700
3097 IF B=4 THEN  GO SUB 3800
3098 IF B=5 THEN  RETURN
3099 GO TO 3000
3100 PAPER 7: CLS
3105 BEEP .2,1
3110 LET X=0: LET Y=0: LET F$(1)
=""
3115 PRINT AT 21,0; FLASH 1;"PRE
SS ENTER TO ACCEPT THE SCREEN"
3130 PRINT AT X,Y; OVER 1;" "
3140 LET A$=INKEY$
3142 IF CODE A$=0 THEN  GO TO 31
30
3143 IF CODE A$=13 THEN  GO TO 3
295
3145 BEEP .05,-60
3147 PRINT AT X,Y;F$(1,X*32+Y+1)
3150 IF CODE A$>20 THEN  PRINT A
T X,Y;" ": BEEP .1,1: PRINT AT X
,Y;A$: LET F$(1,X*32+Y+1)=A$
3155 IF CODE A$=6 AND PEEK 23658
=0 THEN  POKE 23658,8: BEEP .1,2
0: PAUSE 0: GO TO 3130
3157 IF CODE A$=6 AND PEEK 23658
=8 THEN  POKE 23658,0: BEEP .1,-
20: PAUSE 0
3160 IF CODE A$=10 AND X<20 THEN
  LET X=X+1
3165 IF CODE A$=11 AND X>0 THEN
 LET X=X-1
3167 IF CODE A$=10 OR CODE A$=11
 THEN  GO TO 3185
3170 IF CODE A$=8 THEN  LET Y=Y-
1: GO TO 3185
3175 IF CODE A$=7 THEN  LET Y=0:
 LET X=X+1: GO TO 3185
3180 LET Y=Y+1
3185 IF Y>31 THEN  LET Y=0: LET
X=X+1
3190 IF Y<0 THEN  LET Y=31: LET
X=X-11
3195 IF X>20 THEN  LET X=20: LET
 Y=31
3200 IF X<0 THEN  LET X=0: LET Y
=0
3290 GO TO 3130
3295 RETURN
3300 BORDER 5: PAPER 7: INK 0: C
LS
3302 IF ND>1 THEN  PRINT AT 10,0
;"YOU ARE ABOUT TO OVERWRITE DAT
A";AT 12,0;"PRESENTLY STORED IN
MEMORY ";AT 14,0;"ENTER 0 TO CON
TINUE OR";AT 16,0;"ENTER 1 TO RE
TURN TO MAIN MENU": GO SUB 3950
3304 IF ND>1 AND SW=1 THEN  LET
SW=0: GO TO 3490
3305 CLS
3310 LET X=0: LET Y=0
3315 LET NT=0: DIM T(40,9)
3316 LET ND=1
3320 PRINT F$(1)
3330 PRINT AT X,Y; OVER 1;" "
3340 LET A$=INKEY$
3350 IF CODE A$=0 THEN  GO TO 33
30
3360 PRINT AT X,Y;F$(1,X*32+Y+1)
3370 BEEP .1,1
3375 IF CODE A$=13 THEN  GO SUB
3500
3380 IF CODE A$=8 THEN  LET Y=Y-
1
3381 IF CODE A$=12 AND NT=0 THEN
  PRINT AT 21,0; FLASH 1;"NO DAT
A FIELDS DECLARED          ": BEE
P 2,1: GO TO 3490
3385 IF CODE A$=12 THEN  GO SUB
4500: GO TO 3490
3390 IF CODE A$=10 AND X<20 THEN
  LET X=X+1
3400 IF CODE A$=11 AND X>0 THEN
 LET X=X-1
3410 IF CODE A$=9 THEN  LET Y=Y+
1
3415 IF CODE A$=7 THEN  LET Y=0:
 LET X=X+1
3420 IF Y>31 THEN  LET Y=0: LET
X=X+1
3430 IF Y<0 THEN  LET Y=31: LET
X=X-1
3440 IF X<0 THEN  LET X=0: LET Y
=0
3450 IF X>20 THEN  LET X=20: LET
```

```
 Y=31
3480 GO TO 3330
3490 RETURN
3500 REM *********************
3502 REM PUT VALUES IN DIM T
3504 REM *********************
3510 LET NT=NT+1
3520 LET T(NT,1)=X
3530 LET T(NT,2)=Y
3540 INPUT "LENGTH OF FIELD IN C
HARACTERS    ";C
3545 IF C<1 OR C>30 THEN  GO TO
3540
3550 LET T(NT,3)=C
3560 INPUT "PAPER COLOUR OF FIEL
D               ";C
3565 IF C<0 OR C>7 THEN  GO TO 3
560
3570 LET T(NT,5)=C
3580 INPUT "INK COLOUR OF FIELD
                ";C
3585 IF C<0 OR C>7 THEN  GO TO 3
580
3590 LET T(NT,6)=C
3595 INPUT "ENTER BRIGHT 0 OR 1
                ";C
3600 IF C<0 OR C>1 THEN  GO TO 3
595
3610 LET T(NT,7)=C
3615 INPUT "ENTER FLASH  0 OR 1
                ";C
3620 IF C<0 OR C>1 THEN  GO TO 3
615
3625 LET T(NT,8)=C
3630 INPUT "ENTER INVERSE  0 OR
1               ";C
3635 IF C<0 OR C>1 THEN  GO TO 3
630
3640 LET T(NT,9)=C
3650 IF NT=1 THEN  LET T(NT,4)=1
3660 IF NT>1 THEN  LET T(NT,4)=T
(NT-1,4)+T(NT-1,3)
3665 LET X1=X: LET Y1=Y
3670 FOR C=1 TO T(NT,3)
3680 PRINT AT X,Y; PAPER T(NT,5)
; INK T(NT,6); BRIGHT T(NT,7); F
LASH T(NT,8); INVERSE T(NT,9);"X
"
3681 LET Y=Y+1: IF Y>31 THEN  LE
T Y=0: LET X=X+1
3682 IF X>21 THEN  LET X=21: LET
 Y=31
3685 NEXT C
3686 INPUT "ENTER 0 TO CLEAR DAT
A IF YOU    MADE A MISTAKE OTHER
WISE ENTER   ";A$
3687 IF A$="" THEN  BEEP .25,1:
GO TO 3699
3688 IF A$<>"0" THEN  BEEP .25,1
: GO TO 3686
3690 LET X=X1: LET Y=Y1
3692 FOR C=1 TO T(NT,3)
3693 PRINT AT X,Y;" "
3694 LET Y=Y+1: IF Y>31 THEN  LE
T Y=0: LET X=X+1
3695 IF X>20 THEN  LET X=20: LET
 Y=31
3696 NEXT C
3697 LET X=X1: LET Y=Y1
3698 LET NT=NT-1
3699 RETURN
3700 BORDER 5: PAPER 7: INK 0: C
LS
3720 PRINT F$(1)
3730 FOR D=1 TO NT
3740 PRINT AT T(D,1),T(D,2); PAP
ER T(D,5); INK T(D,6); BRIGHT T(
D,7); FLASH T(D,8); INVERSE T(D,
9);X$(1 TO T(D,3))
3760 NEXT D
3770 PRINT AT 21,0; FLASH 1;"PRE
SS ANY KEY TO CONTINUE"
3790 PAUSE 0
3795 RETURN
3800 PAPER 7: INK 0: CLS
3805 LET X=1
3810 PRINT AT 0,5;"RAW DATA DEFI
NITION"
3815 PRINT AT 1,0;"ROW/COL LENGT
H START ADDR. COLOR"
3820 FOR D=1 TO NT
3830 LET X=X+1
3840 IF X>20 THEN  LET X=1: PRIN
T AT 21,0; FLASH 1;"PRESS ANY KE
Y TO CONTINUE": PAUSE 0: PRINT A
T 2,0;,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,: GO TO 3890
3850 PRINT AT X,0;T(D,1);TAB 4;T
(D,2);TAB 10;T(D,3);TAB 18;T(D,4
);TAB 27;T(D,5);T(D,6);T(D,7);T(
D,8);T(D,9)
3890 NEXT D
3895 PRINT AT 21,0; FLASH 1;"PRE
SS ANY KEY TO CONTINUE": PAUSE 0
3900 RETURN
3950 LET A$=INKEY$
3955 IF CODE A$=0 THEN  GO TO 39
50
3960 IF CODE A$=48 THEN  LET SW=
0: GO TO 3990
3965 IF CODE A$=49 THEN  LET SW=
1: GO TO 3990
3970 GO TO 3950
3990 RETURN
4000 REM *********************
4002 REM SAVE AND LOAD ROUTINES
4004 REM *********************
4010 GO SUB 4800
4020 PRINT AT 10,2; FLASH 1;"THE
 FORMAT ARRAY FOLLOWED BY";AT 11
,2;"THE TABLE AND DATA ARRAY
";AT 12,2;"WILL NOW BE SAVED
```

```
      "
4030 INPUT "INPUT FORMAT NAME < 
9 CHARS LONGAND THEN PRESS ENTER
          ";A$
4034 IF LEN A$<1 THEN  GO TO 403
0
4035 IF LEN A$>8 THEN  GO TO 403
0
4040 SAVE A$ DATA F$()
4050 LET A$=A$+"-T"
4060 SAVE A$ DATA T()
4070 LET A$(LEN A$ TO LEN A$)="D
"
4080 SAVE A$ DATA D$()
4190 RETURN
4200 GO SUB 4800
4220 PRINT AT 10,2; FLASH 1;"THE
 FORMAT ARRAY FOLLOWED BY";AT 11
,2;"THE TABLE AND DATA ARRAY
";AT 12,2;"WILL NOW BE LOADED
      "
4230 INPUT "INPUT FORMAT NAME AN
D ENTER     ";"OR JUST PRESS ENT
ER TO LOAD     ";"ANY FORMAT
                ";A$
4235 DIM T(40,9)
4240 LOAD A$ DATA F$()
4250 LET A$=A$+"-T"
4255 IF LEN A$<3 THEN  LET A$=""
4260 LOAD A$ DATA T()
4270 GO SUB 4600
4280 GO SUB 4500
4290 IF LEN A$<3 THEN  GO TO 431
0
4300 LET A$(LEN A$ TO LEN A$)="D
"
4310 LOAD A$ DATA D$()
4320 GO SUB 4700
4390 RETURN
4500 REM ***********************
4502 REM SET DIM D$ SIZE
4504 REM ***********************
4510 LET DIMY=T(NT,4)+T(NT,3)-1
4520 LET DIMX=INT (16000/DIMY)
4530 DIM D$(DIMX,DIMY)
4590 RETURN
4600 REM ************************
4602 REM CALC NO OF FIELDS IN T
4604 REM ************************
4610 FOR D=1 TO 50
4620 IF T(D,3)=0 THEN  LET NT=D-
1: GO TO 4690
4630 NEXT D
4640 PRINT AT 21,0; FLASH 1;"ERR
OR IN CALC OF NT - ABORT RUN ":
STOP
4690 RETURN
4700 REM ************************
4702 REM CALC NO OF RECS IN D$
4704 REM ************************
4710 DIM S$(1,DIMY)
4720 LET S$(1)=""
4730 FOR D=1 TO DIMX
4740 IF D$(D)=S$(1) THEN  LET ND
=D: GO TO 4790
4750 NEXT D
4760 PRINT AT 21,0; FLASH 1;"ERR
OR IN CALC OF ND - ABORT RUN ":
STOP
4790 RETURN
4800 REM ***********************
4802 REM COMMON SUBROUTINE
4804 REM ***********************
4810 BORDER 6: PAPER 6: INK 0
4811 IF A=1 THEN  BORDER 2: PAPE
R 2: INK 7
4812 IF A=2 THEN  BORDER 5: PAPE
R 5: INK 0
4815 IF A=8 THEN  BORDER 1: PAPE
R 1: INK 7
4816 IF A=0 THEN  BORDER 4: PAPE
R 4: INK 0
4820 CLS
4830 PLOT 40,175: DRAW -40,0:
 DRAW 0,-151: DRAW 255,0:
  DRAW 0,151:  DRAW -48,0
4835 IF A=1 THEN  PRINT AT 0,5;"
 INSTRUCTIONS & SCOPE  ";AT 1,5;
"++++++++++++++++++++"
4840 IF A=2 THEN  PRINT AT 0,5;"
 FORMAT SCREEN MENU  ";AT 1,5;"+
++++++++++++++++++"
4860 IF A=6 THEN  PRINT AT 0,5;"
 SAVE FORMAT AND DATA ";AT 1,5;"
++++++++++++++++++++"
4870 IF A=7 THEN  PRINT AT 0,5;"
 LOAD FORMAT AND DATA ";AT 1,5;"
++++++++++++++++++++"
4880 IF A=8 THEN  PRINT AT 0,5;"
USER DEFINED PROGRAM ";AT 1,5;"+
++++++++++++++++++"
4890 IF A=0 THEN  PRINT AT 0,5;"
--END CHECK ROUTINE-- ";AT 1,5;"
++++++++++++++++++++"
4990 RETURN
5000 REM ************************
5002 REM DATA CAPTURE SECTION
5004 REM ************************
5010 PAPER 7: INK 0: BORDER 5: C
LS
5012 IF NT<2 THEN  PRINT AT 21,0
;"NO FORMAT FOR DATA CATURE EXIS
TS": BEEP 3,1: GO TO 5990
5020 LET SW=0
5030 IF SW=1 THEN  LET SW=0: GO
TO 5990
5040 PRINT AT 0,0;F$(1)
5050 PRINT AT 21,0;"BYTES USED "
;ND*DIMY;AT 21,16;" AVAILABLE ";
DIMX*DIMY
5060 LET PT=1: LET C=0: LET D=1
5070 LET X=T(PT,1): LET Y=T(PT,2
```

```
)
5100 PRINT AT X,Y; OVER 1;" "
5110 LET A$=INKEY$
5120 IF CODE A$=0 THEN  GO TO 51
10
5125 IF PEEK 23658=8 THEN  BEEP
.01,20
5127 IF PEEK 23658=0 THEN  BEEP
.01,-10
5130 IF CODE A$>20 THEN  LET C=C
+1: LET D=D+1: PRINT AT X,Y;" ":
 PRINT AT X,Y; PAPER T(PT,5); IN
K T(PT,6); BRIGHT T(PT,7); FLASH
 T(PT,8); INVERSE T(PT,9);A$: LE
T D$(ND,C)=A$: LET Y=Y+1: GO SUB
 5400
5140 IF CODE A$=8 THEN  GO SUB 5
600: LET C=C-1: LET D=D-1: LET Y
=Y-1: GO SUB 5400
5150 IF CODE A$=9 THEN  GO SUB 5
600: LET C=C+1: LET D=D+1: LET Y
=Y+1: GO SUB 5400
5160 IF CODE A$=10 THEN  GO SUB
5600: LET PT=PT+1: GO SUB 5500
5170 IF CODE A$=11 THEN  GO SUB
5600: LET PT=PT-1: GO SUB 5500
5180 IF CODE A$=6 AND PEEK 23658
=0 THEN  POKE 23658,8: BEEP .1,2
0: PAUSE 0: GO TO 5100
5190 IF CODE A$=6 AND PEEK 23658
=8 THEN  POKE 23658,0: BEEP .1,-
20: PAUSE 0
5200 IF CODE A$=13 THEN  PRINT A
T 21,0;,,: PRINT AT 21,0; FLASH
1;"DOC. NO ";ND;" STORED   ": LE
T ND=ND+1: BEEP 2,1
5210 IF CODE A$=13 AND A=4 THEN
 GO SUB 5700: GO TO 5990
5220 IF CODE A$=13 THEN  GO SUB
5700: GO TO 5030
5250 GO TO 5100
5400 IF D>T(PT,3) THEN  LET PT=P
T+1: GO SUB 5500
5410 IF D<1 THEN  LET PT=PT-1: G
O SUB 5500
5420 IF Y>31 AND X<20 THEN  LET
Y=0: LET X=X+1
5430 IF Y<0 AND X>0 THEN  LET Y=
31: LET X=X-1
5490 RETURN
5500 IF PT<1 THEN  LET PT=1
5510 IF PT>NT THEN  LET PT=NT
5520 LET X=T(PT,1): LET Y=T(PT,2
)
5530 LET C=T(PT,4)-1
5540 LET D=1
5590 RETURN
5600 PRINT AT X,Y; PAPER T(PT,5)
; INK T(PT,6); BRIGHT T(PT,7); F
LASH T(PT,8); INVERSE T(PT,9);D$
(ND,C+1)
5690 RETURN
```

```
5700 CLS
5705 IF A=4 THEN  LET SW=1: GO T
O 5790
5710 PRINT AT 20,0; FLASH 1;"ENT
ER 0 TO CON'T DATA CAPTURE   ";A
T 21,0; FLASH 1;"ENTER 1 TO RETU
RN TO MAIN MENU  "
5720 LET A$=INKEY$
5730 IF CODE A$=0 THEN  GO TO 57
20
5740 IF CODE A$=48 THEN  LET SW=
0: CLS : GO TO 5790
5750 IF CODE A$=49 THEN  LET SW=
1: CLS : GO TO 5790
5760 GO TO 5720
5790 RETURN
5990 RETURN
6000 REM ************************
6002 REM ENQUIRY/EDIT ROUTINES
6004 REM ************************
6010 GO SUB 9200
6020 IF ND<2 THEN  GO TO 6990
6040 PRINT AT 21,0; FLASH 1;"INP
UT CHAR. STRING TO SEARCH FOR"
6050 INPUT ;B$
6055 LET F=LEN B$
6060 IF F>T(C,3) OR F<1 THEN  PR
INT AT 21,0; FLASH 1;"INVALID FI
ELD LENGTH - RETYPE   ": BEEP 2,
1: PRINT AT 21,0;,,: GO TO 6040
6065 PRINT AT 21,0;,,
6070 FOR E=1 TO ND
6075 IF E>ND THEN  GO TO 6160
6080 PRINT AT 21,0; FLASH 1;"SEA
RCHING DOCUMENT NO. ";E
6090 FOR D=T(C,4) TO T(C,4)+T(C,
3)-F
6100 IF B$=D$(E,D TO D+F-1) THEN
  LET SW=1
6110 NEXT D
6120 IF SW=1 THEN  GO SUB 6300:
  LET SW=0
6130 IF SW1=9 THEN  LET SW1=0: G
O TO 6160
6150 NEXT E
6160 PRINT AT 21,0; FLASH 1;"END
 OF DATA SEARCH ENCOUNTERED  ":
BEEP 3,1
6190 RETURN
6300 CLS
6305 PRINT AT 0,0;F$(1)
6310 FOR G=1 TO NT
6320 LET X=T(G,1): LET Y=T(G,2)
6360 PRINT AT X,Y; PAPER T(G,5);
 INK T(G,6); BRIGHT T(G,7); FLAS
H T(G,8); INVERSE T(G,9);D$(E,T(
G,4) TO T(G,4)+T(G,3)-1)
6380 NEXT G
6390 PRINT AT 21,0; FLASH 1;"1-A
LT 2-DEL 3-PRT 4-CON'T 5-QUIT"
6400 LET A$=INKEY$
6410 IF CODE A$=0 THEN  GO TO 64
```

```
ØØ
6415 PRINT AT 21,Ø;,,
6420 IF CODE A$<49 OR CODE A$>53
 THEN  GO TO 6390
6430 IF CODE A$=49 THEN  GO SUB
 6500: GO TO 6300
6440 IF CODE A$=50 THEN : GO SUB
 6700: GO TO 6990
6450 IF CODE A$=51 THEN  PRINT A
T 21,Ø;,,: COPY : GO TO 6390
6460 IF CODE A$=52 THEN  GO TO 6
990
6470 IF CODE A$=53 THEN  LET SW1
=9: GO TO 6990
6490 GO TO 6390
6500 REM **********************
6502 REM CHANGE SUBROUTINE
6504 REM **********************
6510 LET H=ND
6520 LET ND=E
6530 LET SW=Ø
6540 LET I=C: LET J=D
6570 GO SUB 5050
6580 LET ND=H
6585 LET C=I: LET D=J
6590 RETURN
6700 REM **********************
6702 REM DELETE SUBROUTINE
6704 REM **********************
6705 IF E=ND THEN  LET D$(ND)=""
: GO TO 6780
6710 PRINT AT 21,Ø; FLASH 1;"WAI
T WHILE I DELETE & COMPACT "
6720 FOR G=E TO ND-1
6730 PRINT AT 21,29;G
6740 LET D$(G)=D$(G+1)
6750 NEXT G
6760 LET D$(ND)=""
6770 LET ND=ND-1
6775 LET E=E-1
6780 PRINT AT 21,Ø;,,
6790 RETURN
6990 RETURN
7000 REM **********************
7002 REM DISPLAY DOCUMENTS
7004 REM **********************
7005 LET SW=Ø
7006 PAPER 7: INK Ø: BORDER 5:
 CLS
7007 IF ND<2 THEN  PRINT AT 21,Ø
;"NO DOCUMENTS TO DISPLAY": BEEP
 3,1: GO TO 7490
7010 FOR C=1 TO ND-1
7015 IF SW=1 THEN  GO SUB 7400
7016 IF SW=9 THEN  LET SW=Ø: GO
TO 7300
7020 PRINT AT Ø,Ø;F$(1)
7030 PRINT AT 21,Ø; FLASH 1;"PRE
SS ANY KEY TO HALT DISPLAY"
7040 FOR D=1 TO NT
7050 LET X=T(D,1): LET Y=T(D,2)
7065 IF INKEY$<>"" THEN  LET SW=
1
7090 PRINT AT X,Y; PAPER T(D,5);
 INK T(D,6); BRIGHT T(D,7); FLAS
H T(D,8); INVERSE T(D,9);D$(C,T(
D,4) TO T(D,4)+T(D,3)-1)
7110 NEXT D
7120 PAUSE 20
7150 NEXT C
7300 RETURN
7400 PRINT AT 21,Ø; FLASH 1;"Ø -
 CONTINUE 1 - GO TO MAIN MENU"
7405 LET SW=Ø
7410 LET A$=INKEY$
7420 IF A$="" THEN  GO TO 7410
7430 IF A$="Ø" THEN  GO TO 7490
7440 IF A$="1" THEN  LET SW=9: G
O TO 7490
7450 GO TO 7410
7490 RETURN
8000 REM **********************
8002 REM TOTAL ACCUMULATION
8004 REM **********************
8005 GO SUB 9200
8007 IF ND<2 THEN  GO TO 8400
8010 LET CB=C
8020 LET A=-8
8030 GO SUB 9200
8035 LET A=8: LET NM=1
8036 DIM M$(40,T(CB,3)): DIM M(4
Ø,2): LET MW=Ø: LET A$="
                        "
8037 PRINT AT 21,Ø; FLASH 1;"WAI
T WHILE I ACCUMULATE DATA     "
8040 FOR D=1 TO ND-1
8045 PRINT AT 21,29;D
8050 LET SW=Ø
8150 FOR E=1 TO NM
8160 IF D$(D,T(CB,4) TO T(CB,4)+
T(CB,3)-1)=M$(E) THEN  LET SW=1:
 GO TO 8200
8190 NEXT E
8200 IF SW=Ø THEN  LET NM=NM+1:
LET M$(NM-1)=D$(D,T(CB,4) TO T(C
B,4)+T(CB,3)-1): LET E=NM-1
8202 IF NM=41 THEN  LET NM=40: L
ET M$(40)=""
8204 IF D$(D,T(C,4) TO T(C,4)+T(
C,3)-1)=A$(1 TO T(C,3)) THEN  PR
INT #Ø;AT 1,Ø;"SPACE FILLED FIEL
D CANNOT ACCUM ": BEEP 2,1: PRIN
T #Ø;AT 1,Ø;,,: GO TO 8300
8206 GO SUB 8500
8207 IF SW=9 THEN  PRINT #Ø;AT 1
,Ø;"NON NUMERIC FIELD IGNORED
    ": BEEP 2,1: PRINT #Ø;AT 1,Ø
;,,: GO TO 8300
8210 LET M(E,1)=M(E,1)+1
8220 LET MW=VAL D$(D,T(C,4) TO T
(C,4)+T(C,3)-1)
8230 LET M(E,2)=M(E,2)+MW
```

```
8300 NEXT D
8304 CLS
8305 PRINT AT 0,0; INVERSE 1;"FI
ELD     NO OF RECS    VALUE      "
8310 FOR E=1 TO NM-1
8315 LET D=E
8316 IF D>20 THEN  LET D=D-20
8320 PRINT AT D,0;M$(E);TAB 15;M
(E,1);TAB 23;M(E,2)
8330 IF E=20 THEN  PRINT #0; FLA
SH 1;"PRESS ANY KEY TO CONTINUE
      "
8340 IF E=20 THEN  PAUSE 0
8350 IF E=20 THEN  PRINT AT 1,0;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;
8390 NEXT E
8395 BEEP 3,1
8396 IF E<20 THEN  PRINT #0;"PRE
SS ANY KEY TO CONTINUE"
8397 PAUSE 0
8400 RETURN
8500 FOR G=0 TO T(C,3)-1
8510 IF CODE D$(D,T(C,4)+G)=32
 THEN  GO TO 8590
8520 IF CODE D$(D,T(C,4)+G)=46
 THEN  GO TO 8590
8530 IF CODE D$(D,T(C,4)+G)>47 A
ND CODE D$(D,T(C,4)+G)<58 THEN
GO TO 8590
8540 LET SW=9: LET G=T(C,3)-1
8590 NEXT G
8600 RETURN
9000 REM ************************
9002 REM END ROUTINES
9004 REM ************************
9010 GO SUB 4800
9020 PRINT AT 10,2;"IF YOU FORGO
T TO SAVE YOUR ";AT 11,2;"DATA T
HEN THIS IS A REMINDER"
9030 PRINT AT 21,0;"ENTER: 0 TO
END   1 TO SAVE DATA"
9040 LET A$=INKEY$
9050 IF CODE A$=0 THEN  GO TO 90
40
9055 IF CODE A$=48 THEN  GO TO 9
060
9056 IF CODE A$=49 THEN  GO TO 9
150
9057 GO TO 9040
9060 CLS
9070 PRINT AT 10,2; INVERSE 1;"
END OF RUN "
9100 STOP
9150 RETURN
9200 REM ************************
9202 REM SORT ROUTINE
9204 REM ************************
9210 PAPER 7: INK 0: BORDER 5: C
LS
9212 IF ND<2 THEN  PRINT AT 21,0
;"NO DATA TO PROCESS": BEEP 3,1:
 GO TO 9600
9220 LET SW=0: DIM S$(1,DIMY)
9230 PRINT AT 0,0;F$(1)
9250 FOR D=1 TO NT
9300 PRINT AT T(D,1),T(D,2); PAP
ER T(D,5); INK T(D,6); BRIGHT T(
D,7); FLASH T(D,8); INVERSE T(D,
9);X$(1 TO T(D,3))
9330 NEXT D
9340 PRINT AT 21,0; FLASH 1;"ENT
ER FIELD NO TO BE SORTED ON "
9345 IF A=4 THEN  PRINT AT 21,0;
 FLASH 1;"ENTER FIELD NO TO BE E
NQUIRED ON"
9346 IF A=-8 THEN  PRINT AT 21,0
; FLASH 1;"ENTER FIELD NO TO BE
TOTALED     "
9347 IF A=8 THEN  PRINT AT 21,0;
 FLASH 1;"ENTER CONTROL BREAK FI
ELD NO     "
9350 INPUT C
9380 IF C<1 OR C>NT THEN  PRINT
AT 21,0; FLASH 1;"INVALID VALUE
-TRY AGAIN       ": BEEP 2,1: GO
TO 9340
9390 LET E=1
9400 FOR D=1 TO T(C,3)
9410 PRINT AT T(C,1),T(C,2)+D-1;
 FLASH 1;"*"
9420 NEXT D
9430 PRINT AT 21,0; FLASH 1;"ENT
ER 0 IF O.K. OR 1 TO CHANGE "
9440 LET A$=INKEY$
9450 IF CODE A$=0 THEN  GO TO 94
40
9460 IF A$="1" THEN  GO TO 9200
9470 IF A$<>"0" THEN  GO TO 9440
9475 PRINT AT 21,0;,,
9476 IF A=4 OR A=8 OR A=-8 THEN
 RETURN
9480 PRINT AT 21,0; FLASH 1;"WAI
T WHILE I SORT   PASS NO: ";E
9490 FOR D=1 TO ND-2
9500 IF D$(D,T(C,4) TO T(C,4)+T(
C,3)-1)>D$(D+1,T(C,4) TO T(C,4)+
T(C,3)-1) THEN  LET S$(1)=D$(D+1
): LET D$(D+1)=D$(D): LET D$(D)=
S$(1): LET SW=1
9530 NEXT D
9550 IF SW=1 THEN  LET SW=0: LET
 E=E+1: GO TO 9475
9560 PRINT AT 21,0;,,
9570 PRINT AT 21,0; FLASH 1;"SOR
T FINISHED IN " ;E;" PASSES"
9580 BEEP 3,1
9600 RETURN
9800 PRINT AT 21,0;(PEEK 23730+2
56*PEEK 23731)-(PEEK 23653+256*P
EEK 23654): STOP
9900 RETURN
```

A nice little version of 'amphibious' fun.

This is a version of Frogger for the 48K Spectrum. It has all of the usual features plus some interesting extra features. You must first cross the busy road then avoid the snake which weaves it's way along the central reservation, cross the river by jumping on to logs, and reach the safety of your lilly pad. If the lilly pad has a butterfly on it then you will score extra points but you must be quick or it will soon fly away.

The program has some interesting features that may be of interest to other programmers. The program is compatible with the Kempston joystick, it has a high score table and it redefines that character set.

The Spectrum uses the system variable CHARS to point to the character set and by making this address point to a location in RAM it is possible to redefine the character set. Lines 9500-9925 poke in the new character set and if you include this subroutine in your own programs then the character set will be redefined.

When the program is RUN there will be a long pause while the character set and user defined graphics are poked into the memory.

HOPPIT

BY ROBERT KIRTLAND

```
    1 REM ************************
          *Underlined characters*
          *are entered in       *
          *GRAPHICS mode.       *
          ************************
    2 CLEAR 63999
    5 DIM h(11): DIM h$(11,10)
    6 LET K=0
    7 GO TO 9500
    8 GO SUB 9000: GO SUB 8000
   10 PAPER 0: INK 6: BORDER 0: C
LS
   20 LET y=21: LET x=15
   40 LET f=0: LET SX=4
   50 LET l=3
   60 LET s=0: LET T=0
   70 PRINT AT 0,0;"SCORE ";AT 0,
15; INK 4;"LIVES NNN"
   80 POKE 23658,8
   90 LET o=0
  100 FOR j=4 TO 11
  110 PRINT AT j,0; PAPER 1;"
                             "
  120 NEXT j
  130 FOR j=13 TO 20
  140 PRINT AT j,0; PAPER 0;"
                             "
  150 NEXT j
  160 PRINT AT 3,0; INK 4;"
                             "
  170 PRINT AT 3,7; INK 3;" ";AT
3,15;" ";AT 3,23;" "
  180 PRINT AT 14,0; INK 7;"- - -
- - - - - - - - - - - - - - "
  190 PRINT AT 16,0; INK 7;"_ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ "
  200 PRINT AT 19,0; INK 7;"- - -
- - - - - - - - - - - - - - "
 1000 LET a$="   LM     LM      L
M      LM   "
 1010 LET b$="  CDCDE     CDE
 CDE          "
 1020 LET c$="FGGGGH     FGGGH
   FGGGH      "
 1030 LET d$="    FGGH     FGGGGH
  FGH   FGGGH"
 1500 PRINT AT 18,0; PAPER 0; INK
 6;a$
 1510 PRINT AT 15,0; PAPER
```

```
3;b$
1520 PRINT AT 9,0; PAPER 1; INK
3;c$
1530 PRINT AT 6,0; PAPER 1; INK
3;d$
1535 PRINT AT 12,SX;" P Q"
1540 IF Y=6 THEN  LET X=X+1: IF
X=32 THEN  GO TO 7000
1550 IF Y=9 THEN  LET X=X-1: IF
X=-1 THEN  GO TO 7000
1600 PRINT AT y,x; PAPER 8; INK
4;"N"
1610 PRINT AT 0,6;s
1700 IF RND<.03 AND NOT O THEN
GO SUB 4000
1710 LET T=T+(T>0)
1720 IF T=20 THEN  GO SUB 4200
2000 LET a$=a$(2 TO 32)+a$(1)
2010 LET b$=b$(32)+b$(1 TO 31)
2020 LET c$=c$(2 TO 32)+c$(1)
2030 LET d$=d$(32)+d$(1 TO 31)
2040 LET SX=SX+1: IF SX=30 THEN
 LET SX=1: PRINT AT 12,30;"  "
2500 LET i$=INKEY$
2505 LET i=IN 31
2510 IF i$="" AND i=0 THEN  GO T
O 2585
2520 PRINT AT y,x; PAPER 8;" "
2525 IF K THEN  GO TO 2550
2530 LET x=x+(I$="8" AND x<31)-(
i$="5" AND x>0)
2540 LET y=y+(3 AND i$="6" AND y
<21)-(3 AND i$="7" AND y>0)
2545 GO TO 2570
2550 LET x=x+(I=1 AND x<31)-(i=2
 AND x>0)
2560 LET y=y+(3 AND i=4 AND y<21
)-(3 AND i=8 AND y>0)
2570 IF (SCREEN$ (y,x)<>" " AND
Y>10) OR (ATTR (y,x)=4) THEN  IF
 Y<>6 AND Y<>9 THEN  GO TO 7000
2575 IF Y=6 OR Y=9 THEN  IF SCRE
EN$ (Y,X)=" " THEN  GO TO 7000
2580 PRINT AT y,x; PAPER 8; INK
4;"N"
2585 IF Y=6 OR Y=9 THEN  GO TO 2
610
2590 IF SCREEN$ (y,x-1)<>" " THE
N  GO TO 7000
2600 IF SCREEN$ (y,x+1)<>" " THE
N  GO TO 7000
2610 IF y=3 THEN  GO TO 2700
2615 IF Y<>12 THEN  GO TO 1500
2620 IF Y=12 AND SCREEN$ (Y,X-1)
<>" " THEN  GO TO 7000
2630 GO TO 1500
2700 LET f=f+1
2705 LET s=s+10
2706 IF X=o THEN  LET O=0: LET T
=0: LET s=s+30
2710 IF f=3 THEN  GO TO 5000
2720 LET y=21: LET x=15
3000 GO TO 1500
```

```
4000 LET O=(INT (RND*3)+1)*8-1
4010 IF ATTR (3,0)=4 THEN  LET O
=0: RETURN
4020 PRINT AT 3,0; INK 3;"O"
4030 LET T=1
4050 RETURN
4210 LET T=0
4220 PRINT AT 3,0; INK 3; PAPER
8;" "
4230 LET O=0
4240 RETURN
5000 PRINT AT 3,0; INK 4;"
"
5005 PRINT AT 3,7; INK 3;" ";AT
3,15;" ";AT 3,23;" "
5010 LET y=21: LET x=15
5030 PRINT AT 0,20+1; INK 4;"N"
5040 LET s=s+20
5100 FOR j=-40 TO 40 STEP 2
5110 BEEP .02,j: BEEP .01,40-ABS
 j
5120 NEXT j
5130 LET F=0
5200 GO TO 1500
7000 LET l=l-1
7005 PRINT AT 0,21+L;" "
7010 BEEP .5,-10
7015 IF y=3 AND l>0 THEN  LET y=
21: LET x=15: GO TO 1500
7020 IF l>0 THEN  PRINT AT y,x;
PAPER 8;" ": LET y=21: LET x=15:
 GO TO 1500
7100 FOR j=1 TO 10
7110 IF s>h(j) THEN  LET t=j: GO
 TO 7300
7120 NEXT j
7130 PAPER 6: CLS
7200 GO TO 7400
7300 INPUT "YOU HAVE A HIGH SCOR
E ENTER YOURNAME ";n$
7301 IF LEN N$>10 THEN  BEEP .5,
10: GO TO 7300
7305 PAPER 6: CLS
7310 FOR j=10 TO t STEP -1
7320 LET h(j+1)=h(j)
7330 LET h$(j+1)=h$(j)
7340 NEXT j
7350 LET h(t)=s
7360 LET h$(t)=n$
7400 PRINT AT 0,9; FLASH 1; INK
2; PAPER 6;"HIGHEST SCORES"
7410 FOR j=1 TO 10
7420 PRINT AT 2+j,9; INK (RND*5)
;h$(j);" ";h(j)
7430 NEXT j
7450 INK 2
7500 PLOT 67,155: DRAW 120,0: DR
AW 0,-90: DRAW -120,0: DRAW 0,90
7900 INPUT "Another game y/n ? "
;y$
7910 IF y$="y" OR y$="Y" THEN  R
```

```
ESTORE : GO TO 10
7920 STOP
8000 PAPER 0: INK 6: CLS : PRINT
 AT 0,13; INK 6; FLASH 1;"HOPPIT
"
8010 PRINT
8020 PRINT " YOU MUST CROSS THE
ROAD AND    RIVER TO RETURN TO Y
OUR LILLY   PAD. IF THE PAD HAS A
 BUTTERFLY,O,ON IT THEN YOU WILL
 SCORE EXTRAPOINTS,BUT YOU MUST
BE QUICK IT WILL ONLY STAY THERE
 FOR A SHORTTIME.ALSO WATCH THE
SNAKE ON THECENTRAL RESERVATION.
"
8030 INPUT "Kempston Joystick ?
";k$
8040 IF k$(1)="y" OR k$(1)="Y" T
HEN  LET k=1
8050 CLS : RETURN
9000 RESTORE 9100: FOR j=USR "a"
 TO USR "Q"+7
9010 READ a
9020 POKE j,a
9030 NEXT j
9040 RETURN
9100 DATA 15,18,34,127,255,255,4
0,16,128,64,32,254,254,255,40,16
9110 DATA 127,127,127,127,127,25
5,21,8,254,254,254,254,255,255,6
4,128
9120 DATA 0,248,196,196,254,254,
40,16,15,12,18,18,18,18,12,15
9130 DATA 255,24,0,60,0,7,0,255,
128,64,32,32,32,32,64,128
9140 DATA 96,124,84,120,127,255,
254,252,0,0,3,2,15,63,255,0,6,12
,152,240,224,85,255,0
9150 DATA 1,2,4,127,127,255,20,8
,240,72,68,254,255,255,20,8
9160 DATA 153,189,255,126,60,189
,165,66
9170 DATA 0,42,93,73,73,42,20,0
9180 DATA 0,0,0,1,67,230,124,56,
0,0,3,195,231,102,60,24
9200 RETURN
9500 FOR j=1 TO 10
9510 LET h$(j)="..........."
9520 NEXT j
9540 RESTORE 9710
9600 FOR z=15616 TO 16384
9605 POKE Z+48384,PEEK Z
9610 NEXT z
9620 FOR f=64128 TO 64207
9630 READ a
9640 POKE f,a
9650 NEXT f
9660 FOR f=64264 TO 64470
9670 READ a
9680 POKE f,a
9690 NEXT f
9695 POKE 23607,249
9700 GO TO 8
9710 DATA 0,126,102,106,114,126,
0
9715 DATA 0,24,120,24,24,24,126,
0
9720 DATA 0,126,2,2,126,96,126,0
9725 DATA 0,126,6,62,6,6,126,0
9730 DATA 0,96,96,100,126,4,4,0
9740 DATA 0,126,96,96,126,2,126,
0
9745 DATA 0,126,96,96,126,98,126
,0
9750 DATA 0,126,6,6,6,6,6,0
9755 DATA 0,126,98,126,98,98,126
,0
9760 DATA 0,126,98,98,126,2,2,0
9800 DATA 0,126,98,98,126,98,98,
0
9805 DATA 0,126,98,124,98,98,126
,0
9810 DATA 0,126,96,96,96,96,126,
0
9815 DATA 0,124,98,98,98,98,124,
0
9820 DATA 0,126,96,124,96,96,126
,0
9825 DATA 0,126,96,124,96,96,96,
0
9830 DATA 0,126,96,96,102,98,126
,0
9835 DATA 0,98,98,126,98,98,98,0
9840 DATA 0,126,24,24,24,24,126,
0
9845 DATA 0,126,24,24,24,24,120,
0
9850 DATA 0,100,104,112,104,100,
98,0
9855 DATA 0,96,96,96,96,96,126,0
9860 DATA 0,126,106,106,106,106,
106,0
9865 DATA 0,126,98,98,98,98,98,0
9870 DATA 0,126,98,98,98,98,126,
0
9875 DATA 0,126,98,98,126,96,96,
0
9880 DATA 0,126,98,98,106,102,12
6,0
9885 DATA 0,126,98,98,126,104,10
2,0
9890 DATA 0,126,96,126,2,2,126,0
9895 DATA 0,126,24,24,24,24,24,0
9900 DATA 0,98,98,98,98,98,126,0
9905 DATA 0,98,98,98,98,52,24,0
9910 DATA 0,106,106,106,106,106,
126,0
9915 DATA 0,102,102,24,102,102,1
02,0
9920 DATA 0,98,98,98,126,8,8,0
9925 DATA 0,126,12,24,48,96,126,
```

Z80 DISASSEMBLER

BY MIKE HENNING

Unravel the mysteries (or at least begin to) of your ZX81's operating system.

I was introduced to computers in 1965 when the US Navy decided that I should be a computer and peripheral repairman. The first computer I worked with was almost seven feet tall and four feet square. The computer had a memory access time of eight microseconds, 32K of memory and an instruction repertoire of 77 function codes. The ZX81 that sits on my desk is faster, has a larger repertoire, and I can carry it around in a shoe box along with a tape recorder and printer.

The bulk of the coding that was done in 1965 was done in low level code or machine language. When diagnostic routines had to be written to fix a piece of equipment they were normally written by the technician using machine code. Low level and machine code gave the programmer a great deal more latitude than high level languages because there were no rules (that is still the case in the use of machine code as you will see later). Since there was a lack of guidelines for coding, programmers developed short cuts in coding such as jumping into the middle of a subroutine to perform only part of the task. Short cuts also conserved the precious resource of memory which was the most expensive component of the system in the 1960s. These non-standard practices allowed programmers to put more program into the same amount of memory than they could with high level languages.

I feel a great deal of frustration at times with the BASIC language because of its slow speed and limited amount of data that can be put in the 16K of usable memory of the ZX81. Even if the user has more than 16K, that is the only portion available for BASIC programs so machine language is required for expanded systems. I am sure that I am not the only ZX81 user who has felt this frustration. The only alternative Sinclair users have when memory space becomes a problem is descoping messages and arrays so programs fit within the rampack. Low level or machine code allows the programmer to put more computational power in the same 16K of memory. That is why many of the program tapes available commercially are written in machine code rather than BASIC and generally interact directly with the operating system.

The only thing that prohibited me from interacting with the operating system using my own machine code was that I did not know what was in the 8K ROM. A couple of years ago when I was using the ZX80, I would dump the contents of the 4K ROM on the screen, write them down, and decode the operations using the Z80 microprocessor repertoire. Before doing that again with the ZX81 I decided it would be much easier if I let the computer help me by at least providing the

instruction mnemonic for the machine code. The program described in this article does just that.

Mnenomically speaking

I will be the first person to admit that obtaining the mnemonic is only a small portion of the task of finding out what the code is doing. But this way the menial work is done by the machine and the programmer can devote his talents to the cerebral task of interpreting the code. The man or men who wrote the operating system for the ZX81 did not have much memory to work with, so decoding the machine code also shows you many of those short cuts that have been used by machine code programmers since the beginning of the computer era.

I suppose that you are asking why you would want to do all this work. Machine code in the ZX81 operates much faster than the equivalent BASIC code. As an example, one of the procedures in the REM statement at the beginning of this program changes all the PRINT statements to LPRINT if the printer output has been selected. The execution time of this routine is less than one second. I had originally done the routine in BASIC but it took almost five minutes to do the same function. Therefore speed and more computational power for each byte of memory are the major advantages of using machine code and interacting with the operating system.

If you know what code exists in the operating system and what it does, you do not need to use valuable cells of memory duplicating functions that already exist. Instead you make a call to the ROM. Not only does this save memory in your program but it also saves you the time of writing the code.

The final advantage of knowing what is in the operating system is that you will have a much better understanding of your equipment. You will be able to interact with your system and feel a real rapport with your system.

Entering the program

This program allows you to display on the screen or print on the personal printer, the mnemonics for the machine code contained anywhere in memory. The format for the output is: address, contents, then mnemonic. The addresses are printed in decimal so it is easier to identify the references to BASIC variables as described in your Users Manual. The contents of memory are printed in octal. That is mostly personal preference on my part but it also makes the identification of some of the Z80 instruction fields easier. The repertoire I have for the Z80 is in binary. Since I have been working with binary coded octal (BCO) for eighteen years, octal was a natural choice for me. The mnemonics come from the repertoire that I use. I have noticed that there are some subtle differences in mnemonics used for the Z80. These differences are not major and there should be no problem deciphering what the mnemonics used by this program mean.

If you are not familiar with the Z80 architecture, it is important for you to get a book on the Z80 microprocessor. The books should describe the processor and define the instruction set. Without this guide, the mnemonics will not make sense because they manipulate registers and flags that are unique to the Z80.

Firstly

The first step in entering this program is to input the BASIC statements. Either execute a NEW command or reset the computer before you start. This is necessary so the checksum feature will work. The program is long so it is a good idea to save it often as you are entering it. Statement 2 is a SAVE command. Do not use this command until the entire program is in and checksummed. The use of the SAVE command is described later in the article. Once all the BASIC code has been entered, the machine code and data table can be put in.

The REM statement at the beginning of the program is loaded with three machine code routines. The listing in Table 1 was output from the program so you could see what the routines do and what the output from the program looks like. The first

16514	136	
16515	052	LD HL, (16396)
16516	014	
16517	100	
16518	001	LD BC, 17400
16519	370	
16520	103	
16521	355	SBC HL,BC
16522	102	
16523	104	LD B,H
16524	115	LD C,L
16525	041	LD HL,17400
16526	370	
16527	103	
16528	076	LD A,245
16529	365	
16530	355	CPIR
16531	261	
16532	300	RET NZ
16533	053	DEC HL
16534	066	LD (HL),225
16535	341	
16536	043	INC HL
16537	030	JR 16530
16538	367	
16539	052	LD HL (16396)
16540	014	
16541	100	
16542	001	LD BC,17400
16543	370	
16544	103	
16545	355	SBC HL,BC
16546	102	
16547	104	LD B,H
16548	115	LD C,L
16549	041	LD HL,17400
16550	370	
16551	103	
16552	076	LD A,225
16553	341	
16554	355	CPIR
16555	261	
16556	300	RET NZ
16557	053	DEC HL
16558	066	LD (HL),245
16559	365	
16560	043	INC HL
16561	030	JR 16554
16562	367	
16563	052	LD HL,(16396)
16564	014	
16565	100	
16566	000	NO-OP
16567	021	LD DE,16515
16568	203	
16569	100	
16570	257	XOR A,A
16571	107	LD B,A
16572	355	SBC HL,DE
16573	122	
16574	353	EX DE,HL
16575	041	LD HL,16515
16576	203	
16577	100	
16578	206	ADD A,(HL)
16579	117	LD C,A
16580	043	INC HL
16581	033	DEC DE
16582	172	LD A,D
16583	203	ADD A,E
16584	310	RET Z
16585	171	LS A,C
16586	030	JR 16578
16587	366	

Table 1

routine changes all of the PRINT statements after the beginning set up statements to LPRINT commands. This routine is called if the answer to the question concerning output is yes. The second routine changes all of the LPRINT commands back to PRINT at the end of the run. The final machine code routine performs a checksum. This allows you to see if you have put the program in just as it appears in this article. Table 2 contains the decimal values that are used to input the machine code. To enter the machine code execute a GOTO 9000. The address that the data are being poked into is displayed. Enter the data from Table 2. After all the data have been input the computer will stop with a 9/9099 displayed.

Table 2

16514 94,42,12,64,1,248,67,237,66,68
16524 77,33,248,67,62,245,237,177,192,43
16534 54,225,35,24,247,42,12,64,1,248
16544 67,237,66,68,77,33,248,67,62,225
16554 237,177,192,43,54,245,35,24,247,42
16564 12,64,0,17,131,64,175,71,237,82
16574 285,33,131,64,134,79,35,27,122,131
16584 200,121,24,246

Table 3

01 NO-OP	32 RRCA	63 HL	94 IM0
02 EX AF	33 RLA	64 AF	95 IM1
03 ,AF'	34 RRA	65 POP	96 IM2
04 DJNZ	35 DAA	66 PUSH	97 LD I,
05 JR	36 CPL	67 OUT	98 LD R,
06 JR NZ	37 SCF	68 IN A	99 LD A,
07 JR Z	38 CCF	69 EX (S	100 LD A,
08 JR NC	39 ADD	70 EX DE	101 RRD
09 JR C	40 ADC	71 DI	102 RID
10 LD	41 SUB	72 EI	103 A
11 ADD	42 SBC	73 P),HL	104 A
12 BC	43 AND	74 ,HL	105 I
13 DE	44 XOR	75 CALL	106 R
14 HL	45 OR	76 RLC	107
15 SP	46 CP	77 RRC	108
16 HALT	47 RET	78 RL	109 LDI
17 B	48 EXX	79 RR	110 LDD
18 C	49 JP (H	80 SLA	111 LDIR
19 D	50 LD SP	81 SRA	112 LDDR
20 E	51 L)	82 ?????	113 CPI
21 H	52 ,HL	83 SRL	114 CPD
22 L	53 NZ	84 BIT	115 CPIR
23 (HL)	54 Z	85 RES	116 CPDR
24 A	55 NC	86 SET	117 INI
25 (BC)	56 C	87 IN	118 IND
26 (DE)	57 PO	88 OUT	119 INIR
27 HL	58 PE	89 SBC	120 INDR
28 A	59 P	90 ADC	121 OUTI
29 INC	60 M	91 NEG	122 OUTD
30 DEC	61 BC	92 RETN	123 OTIR
31 RLCA	62 DE	93 RETI	124 OTDR

The final step is to put the data in the mnemonic table. The data in Table 3 are used for this. The numbers in Table 3 are there as a guide only and should not be input. Each item of the mnemonic table is five characters long but it is not necessary to enter all five characters — if the item is less than five, the computer takes care of that. Please notice that items 107 and 108 are blank. This is intentional and necessary. When you input these two items just hit enter. To put the mnemonic table into the program execute a GOTO 9100. The item number of the data element in the table that you are currently putting in will be displayed on the screen. When all of the data has been input, the computer will stop with a 9/9199 displayed.

And now

The program is now complete and should be saved. You are now ready to see if all the BASIC and machine code have been input correctly. (This checksum does not check the data area where the mnemonic table is stored). To perform the checksum execute a GOTO 9200. If the program has been input as it appears in the article, the program will stop with a message that says "CHECKSUM OK". If the checksum is wrong, the message will say "CHECKSUM WRONG". There is no easy way to find out what has been input incorrectly. The process requires a line by line check. For example, the checksum checks all of the statements, therefore, it is possible an incorrect line number is causing the checksum error.

There is no reason why you cannot change the number base or format of the output to suit your particular references, but I would suggest that the program be put in initially just as it appears in the article. That way before you begin any changes you will know that you are working from a valid baseline.

The program does not use all of the 6K RAM. There are about 3000 bytes left. Therefore you can add special features to the program to print specific types of data. I have included three special features in the base program. The first special feature prints the Sinclair character represented by the contents of memory. This feature is used to output memory locations 126 through 507. This area contains the keyboard and keyword tables. The second special feature prints the character pixel data the Sinclair workhorse uses for the display. This table is located at addresses 7680 through 8191. The third special feature will take consecutive memory locations and print the decimal value for n and n+1. This feature is helpful for printing jump tables that are located throughout the system ROM.

Program operation

To operate the program never use RUN. The RUN command does a clear before it begins and this would cause the mnemonic table to be destroyed. In order to alleviate that from accidentally happening, there is a SAVE statement at line 2. When the program has been verified for correctness, a GOTO 2 should be

executed. This will save the program so that it will start execution at statement 3 immediately after loading without operation intervention. When the program is saved in this manner the last character before the final quote mark in the SAVE statement will be displayed in inverse video. The operating system recognises this at load time as a program that is scheduled for automatic execution.

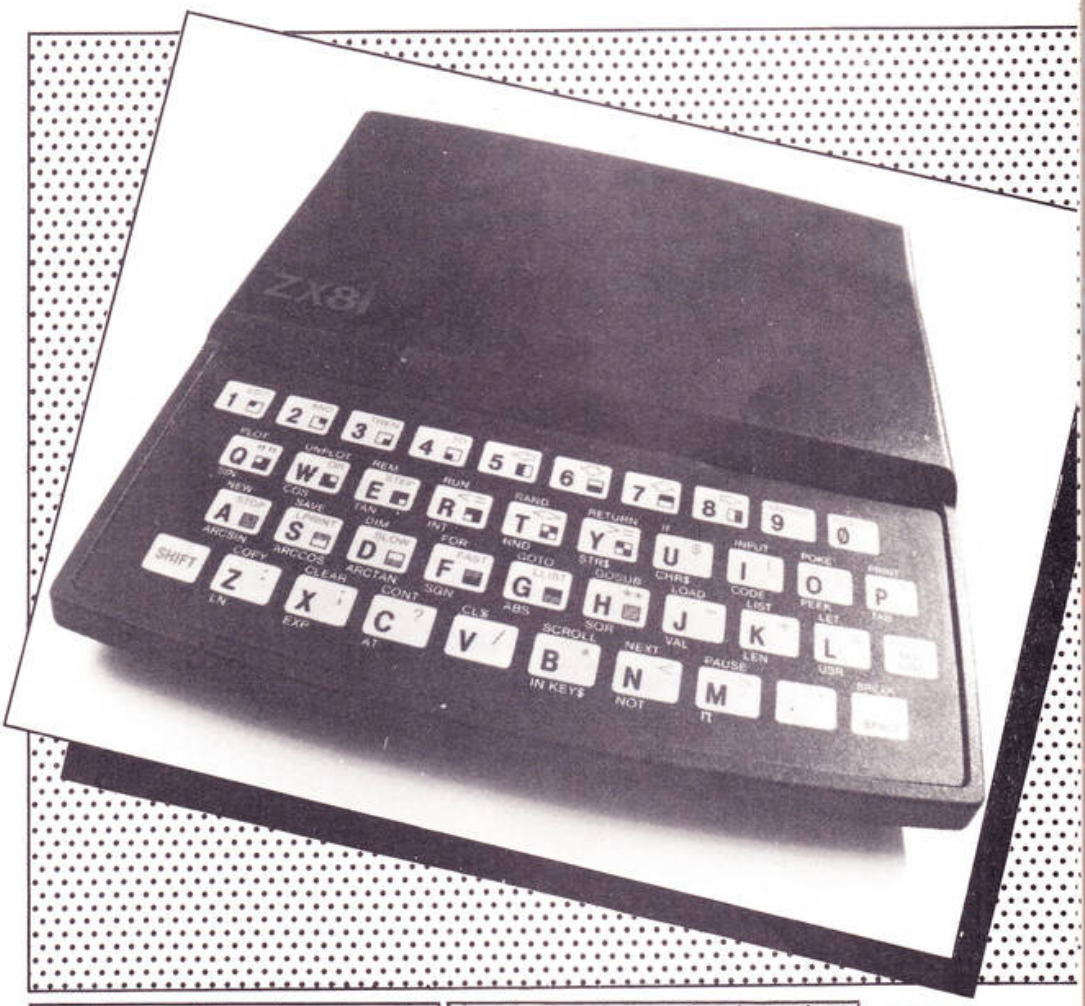
At execution time the first question asked is if you want a special feature. If the answer is "Y", then the menu of special features is presented. A special feature can be selected and the program continues. If the answer is "N" then the special feature menu is not presented and the program will be in the disassemble mode. The next input is the address that you want to start decoding at. There are no validity checks on this input so any area of memory can be decoded. The next question concerns an offset value. If there is memory available in your system in an area that is not used by the BASIC operating system, the contents of machine code located anywhere in the system can be moved to that area and output by the program. The offset allows the computed address of relative jump instructions to be computed correctly. For example, if the contents of ROM were moved to the 8K — 16K block of memory the offset value would be 8192. If the contents of the 8K block starting at 24576 were moved to the same location, the offset value would be −16384. The offset value is the difference between where the program is located for disassembly and the runtime execution location. Printing the system ROM starting at address 0 requires an offset value of 0. A zero for the offset value should be entered anytime the program being disassembled is located at its runtime location. The program continues after entering the offset by asking for the ending address. This input tells the program how far you want to translate. If the ending address is in the midst of a multiple byte instruction, the entire instruction will be output before the program quits. The final operator input is whether you want the output printed on the personal printer. If the answer to the question is "Y" then all of the PRINT statements that output the mnemonics are changed to LPRINT so the output will go to the printer. If the answer is "N" then the output is directed to the screen and the program runs in the SLOW mode and uses SCROLL. When the program completes the requested action the end is signified by "RUN DONE" appearing on the screen. At this point the program is in an idle state executing the PAUSE statement, line 61. By depressing any key other than the space key, the program will continue at line 3. The space key will cause a D type stop.

Don't worry

As the program works through the contents of ROM there will be times that the output is wrong. This is not a program bug!

Because of the limited space that the writers of the operating system had, there are data stored throughout the operating system procedures. There is no way for the program to distinguish between data and instructions so everything is interpreted as an instruction. If the last byte of data contains the code for a multiple byte instruction the first part of the following routine will be in error. It does not take long for the program to get back on track, but be aware that this does occur as you are deciphering your output.

There are a number of special features that can be added to the program. If you have extra memory you could put in a label table, and add labels to your output. Another feature if you have extra memory is a cross reference listing. I am looking forward to seeing more programs that interact with the operating system.



I have not found any bugs in the program that appears in the article but that is not to say that there are none. If you have a problem or a question, I can be reached at the following address: Mike Henning, P.O. Box 2155, Arlington, VA, USA 22202.

```
   1 REM ?E£RND"  SAVE ? GOSUB PI
??5 SAVE ?Y PRINT GOSUB ■""F0 LP
RINT 7/ RUN'E£RND"  SAVE ? GOSUB
PI???5 SAVE ?Y LPRINT GOSUB ■""F0
 PRINT 7/ RUN E£RND ) ■RND■? GOSU
B ? FOR S■RND"■??.?■COS ?/ PLOT
   2 SAVE "DISASSEMBLE"
   3 CLS
   4 PRINT "DO YOU WANT A SPECIA
L FEATURE   (Y,N)?"
   5 INPUT Q$
   6 IF Q$="Y" THEN GOSUB 600
   7 CLS
  10 PRINT "ENTER START ADDRESS,
"
  12 INPUT CELL
  14 PRINT CELL
  16 PRINT AT 4,0;"ENTER ADDRESS
 OFFSET."
  18 INPUT OFFSET
  20 PRINT OFFSET
  22 PRINT AT 7,0;"ENTER ENDING
ADDRESS."
  24 INPUT END
  26 PRINT END
  28 PRINT AT 10,0;"PRINTER OUTP
UT (Y,N)."
  30 INPUT P$
  31 IF P$<>"Y" THEN GOTO 36
  32 LET T1=USR (16514)
  33 PRINT "YES"
  34 FAST
  35 GOTO 40
  36 SLOW
  38 PRINT "NO"
  42 FOR X=CELL TO END
  43 DIM B$(1,10)
  44 GOSUB 1056
  45 IF Q$<>"N" THEN GOTO 680
  46 IF D=203 THEN GOTO 200
  48 IF D=237 THEN GOTO 300
  50 IF D=221 OR D=253 THEN GOTO
 400
  52 GOTO 100
  54 IF CELL>END THEN GOTO 56
  55 NEXT X
  56 IF P$<>"Y" THEN GOTO 63
  57 LET T1=USR (16539)
  58 LPRINT
  59 CLS
  60 PRINT AT 11,10;"RUN DONE"
  61 PAUSE 40000
  62 GOTO 3
  63 SCROLL
  68 PRINT TAB 10;"RUN DONE"
  70 GOTO 61
 100 REM FOR 000-377
 102 IF D76<>0 THEN GOTO 138
 104 GOTO 106+D210*4
 106 GOSUB 2000
 108 GOTO 54
 110 GOSUB 2040
 112 GOTO 54
 114 GOSUB 2074
 116 GOTO 54
 118 GOSUB 2114
 120 GOTO 54
 122 GOSUB 2126
 124 GOTO 54
 126 GOSUB 2138
 128 GOTO 54
 130 GOSUB 2150
 132 GOTO 54
 134 GOSUB 2162
 136 GOTO 54
 138 IF D76<>1 THEN GOTO 144
 140 GOSUB 2300
 142 GOTO 54
 144 IF D76<>2 THEN GOTO 150
 146 GOSUB 2200
 148 GOTO 54
 150 GOTO 152+D210*4
 152 GOSUB 2400
 154 GOTO 54
 156 GOSUB 2412
 158 GOTO 54
 160 GOSUB 2438
 162 GOTO 54
 164 GOSUB 2454
 166 GOTO 54
 168 GOSUB 2492
 170 GOTO 54
 172 GOSUB 2508
 174 GOTO 54
 176 GOSUB 2530
 178 GOTO 54
 180 GOSUB 2544
 182 GOTO 54
 200 REM FOR IX 000-377
 201 GOSUB 700
 202 IF D76<>0 THEN GOTO 208
 204 GOSUB 2600
 206 GOTO 54
 208 GOSUB 2614
 210 GOTO 54
 300 REM FOR IY 000-377
 301 GOSUB 700
 302 IF D76<>1 THEN GOTO 322
 304 IF D210<>1 THEN GOTO 310
 306 GOSUB 2700
 308 GOTO 54
 310 IF D210<>2 THEN GOTO 316
 312 GOSUB 2720
 314 GOTO 54
 316 IF D210<>3 THEN GOTO 320
 318 GOSUB 2732
 319 GOTO 54
 320 GOSUB 2752
 321 GOTO 54
 322 IF D543<4 OR (D543>=4 AND D
210>=4) THEN GOTO 328
 324 GOSUB 2776
 326 GOTO 54
 328 PRINT "■■■■■■■■■■■■■■"
 330 GOSUB 1056
 332 GOTO 54
 400 REM FOR IX IY 000-377
 402 LET T1=D
 406 GOSUB 700
 410 GOSUB 2800
 412 GOTO 54
 600 REM SPECIAL FEATURE
 601 CLS
 602 PRINT "SPECIAL FEATURES"
 604 PRINT
 606 PRINT "1 PRINT TS 1000 CODE
"
 608 PRINT "2 PRINT PIXEL INFO"
 610 PRINT "3 PRINT DNN VALUE"
 640 PRINT AT 19,0;"ENTER FUNCTI
ON"
 642 INPUT Q$
 643 LET T2=1
 644 RETURN
 680 IF Q$="1" THEN GOSUB 3000
 682 IF Q$="2" THEN GOSUB 3100
 684 IF Q$="3" THEN GOSUB 3200
 699 GOTO 54
 700 REM GET NEXT CELL
 704 GOSUB 1000
 708 RETURN
1000 REM GET CELL
1002 DIM D(8)
1003 LET A=PEEK CELL
1004 FOR I=1 TO 8
```

```
1006 LET A1=A/2
1008 LET D(I)=A-INT A1*2
1010 LET A=INT A1
1012 IF A=0 THEN GOTO 1015
1014 NEXT I
1015 LET D=PEEK CELL
1016 LET D76=(D(8)*2)+D(7)
1018 LET D543=(D(6)*4)+(D(5)*2)+
D(4)
1020 LET D210=(D(3)*4)+(D(2)*2)+
D(1)
1022 LET D3=D(4)
1023 LET D43=D(5)*2+D(4)
1024 LET D54=D(6)*2+D(5)
1025 LET D7=D(8)
1026 LET D6T00=(D(7)*64)+(D543*8
)+D210
1028 LET DN=PEEK (CELL+1)
1030 LET DNN=PEEK (CELL+2)*256+P
EEK (CELL+1)
1032 RETURN
1034 REM PRINT ADDRESS
1035 LET B$(1)=STR$ ((CELL-OFFSE
T)/100000)
1036 IF P$="Y" THEN PRINT
1037 IF P$="Y" THEN GOTO 1039
1038 SCROLL
1039 LET T1=1
1040 IF CELL-OFFSET>9999 THEN LE
T T1=2
1041 FOR I=1 TO 5
1042 IF B$(1,I+T1)=" " THEN LET
B$(1,I+T1)="0"
1043 PRINT B$(1,I+T1);
1046 NEXT I
1048 RETURN
1050 REM PRINT CONTENTS OF CELL
1052 PRINT TAB 7;D76;D543;D210;"
";
1054 RETURN
1056 REM SET CELL,PRINT CONTENTS
      ,ADDRESS AND INCREMENT
1059 GOSUB 1000
1060 GOSUB 1034
1062 GOSUB 1050
1064 LET CELL=CELL+1
1065 RETURN
1066 REM PRINT MNEMONIC
1068 FOR I=1 TO 6
1070 IF (I=6 AND B$(1,6)=" ") OR
 (B$(1,I)=" " AND B$(1,I+1)=" ")
 THEN RETURN
1072 PRINT B$(1,I);
1074 NEXT I
1076 RETURN
2000 REM FOR 0?3
2002 IF D543>1 THEN GOTO 2014
2004 LET B$(1)=A$(D543+1)
2006 GOSUB 1066
2008 IF D3<>1 THEN GOTO 2012
2010 LET B$(1)=A$(3)
2011 GOSUB 1066
2012 GOTO 2032
2014 LET B$(1)=A$(D543+2)
2016 GOSUB 1066
2024 IF DN<128 THEN LET A=CELL-O
FFSET+1+DN
2026 IF DN>=128 THEN LET A=CELL-
OFFSET+1-(256-DN)
2028 PRINT " ";A;
2030 GOSUB 1056
2032 RETURN
2040 REM FOR 0?4
2042 IF D3=1 THEN GOTO 2066
2044 LET B$(1)=A$(10)
2046 LET B$(1,3)=" "
2048 LET B$(1,4 TO 5)=A$(D54+12)
2050 GOSUB 1066
2058 PRINT ",";DNN;
2060 GOSUB 1056
2062 GOSUB 1056
2064 RETURN
2066 LET B$(1)=A$(11)
2068 GOSUB 1066
2070 PRINT " ";A$(14,1 TO 2);",
";A$(D54+12);
2072 GOTO 2064
2074 REM FOR 0?5
2076 LET B$(1)=A$(10)
2078 GOSUB 1066
2080 IF D54>1 THEN GOTO 2092
2082 IF D3=1 THEN GOTO 2088
2084 PRINT " ";A$(D54+25,1 TO 4)
;",A";
2086 GOTO 2110
2088 PRINT " A,";A$(D54+25);
2090 GOTO 2110
2092 IF D3=1 THEN GOTO 2098
2094 PRINT " (";DNN;"),";A$(D54+
25);
2096 GOTO 2106
2098 LET B$(1,1)=" "
2100 LET B$(1,2 TO 6)=A$(D54+25)
2102 GOSUB 1066
2104 PRINT ",(";DNN;")";
2106 GOSUB 1056
2108 GOSUB 1056
2112 RETURN
2114 REM FOR 0?6
2116 LET B$(1)=A$(D3+29)
2118 GOSUB 1066
2120 PRINT " ";A$(D54+12);
2124 RETURN
2126 REM FOR 0?7
2128 LET B$(1)=A$(29)
2130 GOSUB 1066
2132 PRINT " ";A$(D543+17);
2136 RETURN
2138 REM FOR 0?8
2140 LET B$(1)=A$(30)
2142 GOSUB 1066
2144 PRINT " ";A$(D543+17);
2148 RETURN
2150 REM FOR 0?9
2152 LET B$(1)=A$(10)
2154 GOSUB 1066
2155 LET B$(1,1)=" "
2156 LET B$(1,2 TO 6)=A$(D543+17
)
2157 GOSUB 1066
2158 PRINT ",";DN;
2160 GOSUB 1056
2161 RETURN
2162 REM FOR 0?A
2164 LET B$(1)=A$(D543+31)
2166 GOSUB 1066
2170 RETURN
2200 REM FOR 0?B
2202 LET B$(1)=A$(D543+39)
2204 GOSUB 1066
2206 PRINT " A,";A$(D210+17);
2210 RETURN
2300 REM FOR 0?C
2302 IF D<>118 THEN GOTO 2310
2304 LET B$(1)=A$(16)
2306 GOSUB 1066
2308 RETURN
2310 LET B$(1)=A$(10)
2311 GOSUB 1066
2312 PRINT " ";
2313 LET B$(1)=A$(D543+17)
2314 GOSUB 1066
2315 PRINT ",";A$(D210+17);
2318 RETURN
2400 REM FOR 0?D
2402 LET B$(1)=A$(47)
2404 GOSUB 1066
2406 PRINT " ";A$(D543+53);
```

```
2410 RETURN
2412 REM FOR 3X
2414 IF D3=1 THEN GOTO 2424
2416 LET B$(1)=A$(65)
2418 GOSUB 1056
2420 PRINT " ";A$(D54+61);
2422 GOTO 2436
2424 LET B$(1)=A$(D54+47)
2426 GOSUB 1056
2428 IF D54<2 THEN GOTO 2436
2430 LET B$(1)=A$(D54+49)
2432 GOSUB 1056
2436 RETURN
2438 REM FOR 3X
2440 LET B$(1)=A$(49,1 TO 2)
2441 LET B$(1,4 TO 5)=A$(D543+53
,1 TO 2)
2442 GOSUB 1056
2444 PRINT ",";DNN;
2446 GOSUB 1056
2448 GOSUB 1056
2452 RETURN
2454 REM FOR 3X
2456 IF D543<>0 THEN GOTO 2468
2458 PRINT "JP ";DNN;
2460 GOSUB 1056
2462 GOSUB 1056
2466 RETURN
2468 IF D543>3 THEN GOTO 2480
2470 LET B$(1)=A$(D543+65)
2472 GOSUB 1056
2474 IF D3=0 THEN PRINT " ";DN;"
,A";
2476 IF D3=1 THEN PRINT ",";DN;
2478 GOTO 2462
2480 LET B$(1)=A$(D543+65)
2482 GOSUB 1056
2484 IF D54<>2 THEN GOTO 2466
2486 LET B$(1)=A$(D3+73)
2488 GOSUB 1056
2490 GOTO 2466
2492 REM FOR 3X
2494 LET B$(1)=A$(75)
2496 GOSUB 1056
2497 LET B$(1)=" "
2498 LET B$(1,2 TO 6)=A$(D543+53
)
2499 GOSUB 1056
2500 PRINT ",";DNN;
2501 GOSUB 1056
2502 GOSUB 1056
2506 RETURN
2508 REM FOR 3X
2510 IF D3=1 THEN GOTO 2522
2512 LET B$(1)=A$(66)
2514 GOSUB 1056
2516 PRINT " ";A$(D54+61)
2520 RETURN
2522 PRINT "CALL ";DNN;
2524 GOSUB 1056
2525 GOSUB 1056
2526 GOTO 2520
2530 LET B$(1)=A$(D543+39)
2534 GOSUB 1056
2536 PRINT " A,";DN;
2538 GOSUB 1056
2542 RETURN
2544 REM FOR 3X
2546 PRINT "RST ";D543*8;
2550 RETURN
2600 REM FOR 3X 0X TO 0X
2602 LET B$(1)=A$(D543+76)
2604 GOSUB 1056
2606 PRINT " ";A$(D210+17)
2608 GOSUB 1056
2612 RETURN
2614 REM FOR 3X 1X TO 3X
2616 LET B$(1)=A$(D76+83)
2618 GOSUB 1056
```

```
2620 PRINT " ";D543;",";A$(D210+
17);
2622 GOSUB 1056
2626 RETURN
2700 REM FOR 3X 1X TO 1X
2702 LET B$(1)=A$(D210+87)
2704 GOSUB 1056
2706 LET B$(1,1)=" "
2708 LET B$(1,2 TO 6)=A$(D543+17
)
2709 GOSUB 1056
2710 PRINT ",(C)";
2711 GOTO 2714
2712 PRINT " (C),";A$(D543+17);
2714 GOSUB 1056
2718 RETURN
2720 REM FOR 3X 1X
2722 LET B$(1)=A$(D3+89)
2724 GOSUB 1056
2726 PRINT " HL,";A$(D54+12);
2728 GOSUB 1056
2730 RETURN
2732 REM FOR 3X 1X
2734 LET B$(1)=A$(10)
2735 GOSUB 1056
2736 IF D3=1 THEN GOTO 2741
2738 PRINT " (";DNN;"),";A$(D54+
12);
2740 GOTO 2745
2741 LET B$(1,1)=" "
2742 LET B$(1,2 TO 6)=A$(D54+12)
2743 GOSUB 1056
2744 PRINT ",(";DNN;")";
2745 GOSUB 1056
2746 GOSUB 1056
2747 GOSUB 1056
2750 RETURN
2752 REM FOR 3X 1X TO 1X
2756 IF D543=0 AND D210=4 THEN L
ET B$(1)=A$(91)
2758 IF D543=0 AND D210=5 THEN L
ET B$(1)=A$(92)
2760 IF D543=1 AND D210=5 THEN L
ET B$(1)=A$(93)
2762 IF D543=0 AND D210=6 THEN L
ET B$(1)=A$(94)
2764 IF D543=2 AND D210=5 THEN L
ET B$(1)=A$(95)
2766 IF D543=3 AND D210=6 THEN L
ET B$(1)=A$(96)
2769 IF D210=7 THEN LET B$(1)=A$
(D543+97)
2770 GOSUB 1056
2772 GOSUB 1056
2774 RETURN
2776 REM FOR 3X 2X
2778 LET B$(1)=A$(D43+109+D210*4
)
2780 GOSUB 1056
2782 GOSUB 1056
2784 RETURN
2800 REM FOR DD IY XXX
2801 LET C$="IY"
2802 IF T1=221 THEN LET C$="IY"
2804 IF D76<>0 OR D210<>1 OR-D3<
>1 THEN GOTO 2810
2806 PRINT "ADD ";C$;",";A$(D54+
12)
2808 GOTO 2936
2810 IF D<>33 THEN GOTO 2816
2812 PRINT "LD ";C$;",";DNN;
2814 GOTO 2932
2816 IF D<>34 THEN GOTO 2822
2818 PRINT "LD (";DNN;"),";C$;
2820 GOTO 2932
2822 IF D<>42 THEN GOTO 2828
2824 PRINT "LD ";C$;",(";DNN;")"
;
2826 GOTO 2932
```

```
2828 IF D<>35 AND D<>43 THEN GOT
O 2838
2830 LET B$(1)=A$(D3+29)
2832 GOSUB 1066
2834 PRINT " ";C$;
2836 GOTO 2936
2838 IF D<>52 AND D<>53 THEN GOT
O 2848
2840 LET B$(1)=A$(D210+25)
2842 GOSUB 1066
2844 PRINT " (";C$;"+";DN;")";
2846 GOTO 2934
2848 IF D<>54 THEN GOTO 2854
2850 PRINT "LD (";C$;"+";DN;"),"
;PEEK (CELL+2);
2852 GOTO 2932
2854 IF D76<>1 OR D543<>6 THEN G
OTO 2860
2856 PRINT "LD (";C$;"+";DN;"),"
;A$(D210+17);
2858 GOTO 2934
2860 IF D<>225 THEN GOTO 2866
2862 PRINT "POP ";C$;
2864 GOTO 2936
2866 IF D<>227 THEN GOTO 2872
2868 PRINT "EX (SP),";C$;
2870 GOTO 2936
2872 IF D<>229 THEN GOTO 2878
2874 PRINT "PUSH ";C$;
2876 GOTO 2936
2878 IF D<>233 THEN GOTO 2884
2880 PRINT "JP (";C$;")";
2882 GOTO 2936
2884 IF D<>249 THEN GOTO 2890
2886 PRINT "LD SP,";C$;
2888 GOTO 2936
2890 IF D76<>1 OR D210<>6 THEN G
OTO 2896
2891 PRINT "LD ";
2892 LIST B$(1)=A$(D543+17)
2893 GOSUB 1066
2894 PRINT ",(";C$;"+";DN;")";
2895 GOTO 2934
2896 IF D76<>2 OR D210<>6 THEN G
OTO 2906
2898 LET B$(1)=A$(D543+39)
2900 GOSUB 1066
2902 PRINT " A,(";C$;"+";DN;")";
2904 GOTO 2934
2906 IF D<>203 THEN GOTO 2938
2908 LET T1=DN
2910 LET CELL=CELL+2
2912 GOSUB 1000
2914 IF D76<>0 THEN GOTO 2924
2916 LET B$(1)=A$(D543+76)
2918 GOSUB 1066
2920 PRINT " (";C$;"+";T1;")";
2922 GOTO 2932
2924 LET B$(1)=A$(D76+83)
2926 GOSUB 1066
2928 PRINT " ";D543;",(";C$;"+";
T1;")";
2930 LET CELL=CELL+2
2932 GOSUB 1056
2934 GOSUB 1056
2936 GOSUB 1056
2940 RETURN
3000 REM PRINT CODE
3002 PRINT CHR$ D;
3006 GOTO 54
3100 REM PRINT PIXELS
3102 FOR I=8 TO 1 STEP -1
3104 IF D(I)=0 THEN PRINT "▒";
3106 IF D(I)=1 THEN PRINT "■";
3108 NEXT I
3109 LET T2=T2+1
3110 IF T2>9 THEN GOTO 3124
3112 LET T2=1
3114 PRINT
```

```
3124 GOTO 54
3200 REM PRINT CHN
3202 PRINT PEEK CELL*256+D;
3204 GOSUB 1056
3208 GOTO 54
9000 REM ENTER MACHINE CODE
9001 FOR X=16515 TO 16587
9002 PRINT AT 3,5;X
9004 INPUT T
9006 POKE X,T
9008 NEXT X
9010 CLS
9099 STOP
9100 REM ENTER MNEMONIC
9101 DIM A$(124,5)
9102 FOR I=1 TO 124
9104 PRINT AT 12,15;I
9106 INPUT A$(I)
9108 NEXT I
9110 CLS
9199 STOP
9200 REM CHECKSUM PROGRAM
9201 LET T=USR 16563
9202 CLS
9204 IF T<>PEEK 16514 THEN PRINT
 "CHECKSUM WRONG"
9206 IF T=PEEK 16514 THEN PRINT
"CHECKSUM OK"
9299 STOP
```

Deflector is a game for the Spectrum 16 or 48K. The aim of the game is to remove the blocks (Graphic A's) from the screen by rolling over them with your ball (Graphic B). The only way to change the direction of your ball is to place "slashes" in its path — these are the division symbol (sym. shift and V) and the back slash (E mode shifted D). Each of these will cause a 90° change of direction. If you get stuck in a loop you can remove all the slashes on the screen by pressing SPACE; this will also re-arrange the positions of the blocks.

The Program

The program is arranged such that every task that the computer undertakes is written in a subroutine of its own. The game loop starts at line 8015 and consists of several GO SUBS and one GO TO which repeats the loop. The UDG loader should be typed in first and then RUN. It should then be NEWed. Line 9900 of the main program should be typed in first to allow saving of the program and the UDG's at the end of each session. The rest of the listing should then be typed in. The program should be saved by "GO TO 9900" as this saves the program to autorun at the correct point and also saves the UDG's with automatic verification.

BY ANDREW SMITH

Removing blocks with a ball sounds easy, doesn't it? Try this game and see just how wrong you can be!

DEFLECTOR

What's Your Score

The reckoning of the score is based on the following formula:

no. of blocks × 10 -(no. of slashes + no. of rubbed out slashes × 5)

Anything over 0 is quite good! The game is difficult when both high numbers of blocks or low numbers are used; the easiest game is in the 25-65 range.

Take note

One problem I encountered and was unable to solve was that after placing a new slash and acting on it a block in the next character position to the slash in the new direction of travel is ignored, however it is picked up on the return journey.

In programming this game I have adopted a "modular" approach, this means that every task that the computer has to undertake is written in a subroutine of its own. The game loop starts at 8015 and consists of several GO SUBS and one GOTO which repeats the loop.

The UDG loader should be typed in first and then RUN, it should then be NEWed. Line 9900 of the main program should be typed in first to allow saving of both the program and UDG's at the end of each session. The rest of the main listing should then be typed in.

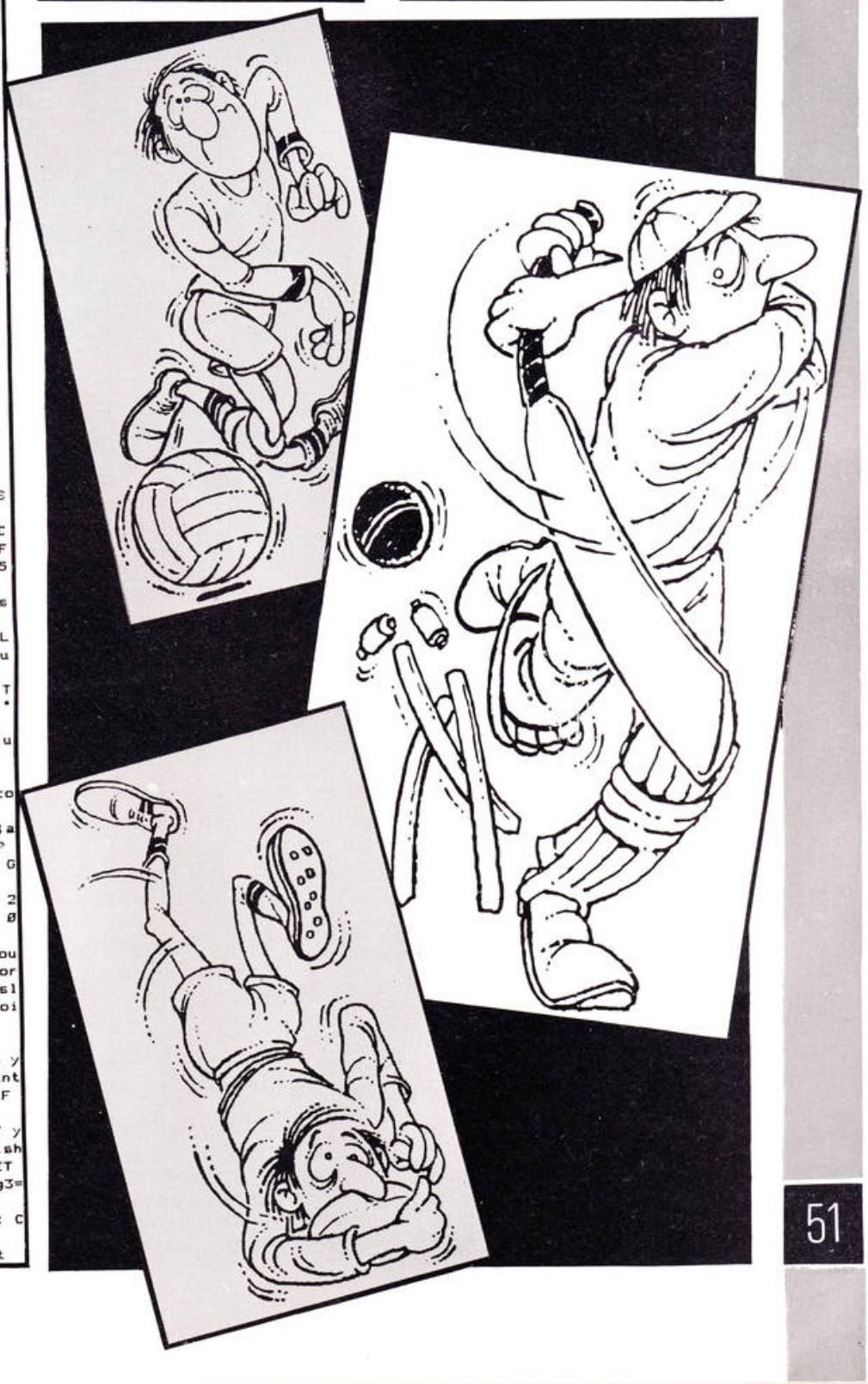
The program should be saved by "GO TO 9900" as this saves the program to autorun at the correct point and also saves the UDGs with automatic verification.

```
   1 GO TO 8000
 100 REM * Slash Check *
 105 IF flag3=1 THEN  RETURN
 110 IF SCREEN$ (y+yd,x+xd)=" "
THEN  RETURN
 115 LET flag1=yd: LET flag2=xd
 117 BEEP .005,50
 120 IF SCREEN$ (y+yd,x+xd)="\"
THEN  GO TO 300
 150 LET z=-yd: LET yd=-xd: LET
xd=z
 160 RETURN
 320 LET z=yd: LET yd=xd: LET xd
=z
 330 RETURN
1000 REM * Move Ball *
1005 LET flag3=0
1010 LET x=x+xd+flag2: LET flag2
=0
1020 LET y=y+yd+flag1: LET flag1
=0
1030 IF x>21 THEN  LET x=21: LET
 xd=-xd: BEEP .005,x+xd+y+yd
1040 IF x<1 THEN  LET x=1: LET x
d=-xd: BEEP .005,x+xd+y+yd
1050 IF y>21 THEN  LET y=21: LET
 yd=-yd: BEEP .005,x+xd+y+yd
1060 IF y<1 THEN  LET y=1: LET y
```

```
=-yd: BEEP .005,x+xd+y+yd
1065 IF y+yd<1 OR y+yd>21 OR x+x
d<1 OR x+xd>21 THEN  LET flag3=1
1070 RETURN
2000 REM * New Slash? *
2001 IF flag3=1 THEN  RETURN
2005 IF INKEY$="" THEN  RETURN
2010 IF INKEY$="q" OR INKEY$="O"
 THEN  LET slshcount=slshcount+1
: PRINT AT y+yd,x+xd;"/"
2020 IF INKEY$="p" OR INKEY$="P"
 THEN  LET slshcount=slshcount+1
: PRINT AT y+yd,x+xd;"\"
2040 RETURN
3000 REM * Block Check *
3005 IF flag3=1 THEN  RETURN
3010 IF ATTR (y+yd,x+xd)<128 THE
N  RETURN
3015 BEEP .008,0: BEEP .008,24
3020 LET blkcount=blkcount-1: LE
T score=score+10
3030 PRINT AT y+yd,x+xd;" "
3040 IF blkcount=0 THEN  GO TO 9
500
3050 RETURN
4000 REM * Escape *
4005 IF INKEY$<>" " THEN  RETURN

4010 CLS
4020 LET esslshcount=esslshcount
+slshcount: LET slshcount=0
4030 GO SUB 9117
4050 RETURN
5000 REM *Print/Unprint Ball*
5010 PRINT AT y,x; OVER 1; FLASH
 0; BRIGHT 0;"●"
5020 RETURN
9000 REM * Main Routine *
9010 GO SUB 9000
9015 GO SUB 1000
9020 GO SUB 5000
9030 GO SUB 3000
9040 GO SUB 2000
9050 GO SUB 100
9060 GO SUB 5000
9070 GO SUB 4000
9080 GO SUB 9200
9090 GO TO 9015
9000 REM * Initialise *
9010 LOAD "UDG"CODE USR "a"
9020 PRINT  BRIGHT 1; FLASH 1;"S
TOP THE TAPE": PAUSE 100
9040 PAPER 7: INK 0: BORDER 7: C
LS : PRINT AT 0,11; PAPER 5;"DEF
LECTOR";AT 0,11; OVER 1; PAPER 5
;"________"'' PAPER 7;""
   Hi there!"''  The aim of this
 game is to       destroy all of
the "; FLASH 1; BRIGHT 1;"■"; FL
ASH 0; BRIGHT 0;" with your  ● u
sing / & \ to control it"
9050 PRINT ''TAB 14;"KEYS"''''  T
o place a / use 'O'          "
''  To place a \ use 'P'
    ''''  To remove the slashes u
se                  (SPACE)
      "
9070 LET a$="Press any key to co
ntinue          "
9080 PRINT #0;AT 0,0; BRIGHT 1;a
$: LET a$=a$(2 TO )+a$(1): BEEP
.03,RND*20: IF INKEY$="" THEN  G
O TO 9080
9090 CLS : PRINT TAB 12; PAPER 2
; BRIGHT 1; INK 7;"WARNING";AT 0
,12; OVER 1;"_______"
9100 PRINT ''''  Every slash you
 use deducts 1     from your scor
e, and every        rubbed out sl
ash deducts                  5 poi
nts"
9110 REM * Place blocks *
9114 INPUT "How many blocks do y
ou wish to  destroy?  ";blkcount
: LET blkcount=ABS blkcount: IF
blkcount>400 THEN  GO TO 9115
9115 LET score=0: LET x=5: LET y
=5: LET xd=1: LET yd=0: LET slsh
count=0: LET esslshcount=0: LET
flag1=0: LET flag2=0: LET flag3=
0
9116 PAPER 5: BORDER 5: INK 1: C
LS
9117 CLS : FOR c=1 TO blkcount
```

```
9120 LET a=INT (RND*21)+1: LET b
=INT (RND*21)+1
9130 IF ATTR (a,b)>128 THEN  GO
TO 9120
9140 PRINT AT a,b; FLASH 1; BRIG
HT 1; INVERSE RND;"■": NEXT c
9170 PRINT AT 0,0;"
";#0;AT 0,0;"
"
9180 FOR a=1 TO 21: PRINT AT a,0
;"▌";AT a,22;"▌": NEXT a
9200 PRINT AT 0,23;"BLOCKS";AT 1
,23;" LEFT";AT 2,24;blkcount;"
9210 PRINT AT 4,23;"SLASHES";AT
5,23;" USED";AT 6,24;slshcount
9300 RETURN
9500 CLS : PRINT AT 9,8; PAPER 7
; BRIGHT 1; FLASH 1;" YOU DID IT
!!!!"
9510 FOR a=1 TO 50: BEEP .008,a:
 BORDER a/7: NEXT a
9520 FOR a=50 TO -50 STEP -1: BE
```

```
EP .008,a: BORDER RND*7: NEXT a
9530  PRINT AT 11,7;"Your score
was ";score-slshcount-5*esslshco
unt
9540 PRINT AT 14,5;"Do you want
another go?": IF INKEY$="" THEN
 GO TO 9540
9550 IF INKEY$="y" OR INKEY$="Y"
 THEN  GO SUB 9114: GO TO 8015
9560 IF INKEY$="n" OR INKEY$="N"
 THEN  GO TO 1e4
9570 GO TO 9540
9900 CLEAR : SAVE "DEFLECTOR" LI
NE 8000: SAVE "UDG"CODE USR "a",
16: VERIFY "DEFLECTOR": VERIFY "
UDG"CODE
   1 REM UDG Loader
   2 FOR a=USR "a" TO USR "c"-1:
 READ b: POKE a,b: NEXT a
   3 DATA 255,129,189,165,165,18
9,129,255,60,126,255,255,255,255
,126,60
```



WRITING DATA

There are a vast number of ways for writing programs to enter and display data using Sinclair BASIC. As a novice programmer with the ambitious goal of writing my own program for a "computerized personnel status board", I have reached some interesting conclusions. The most important is that the manipulation of a large number of rows and columns is no easy task. However, I do have some tips for anyone wanting to write a data base oriented program. First, it is advantageous to group PRINT, INPUT, and DIM statements, for example:

```
10 PRINT A$(N);
20 PRINT B$(N);
      .
      .
      .
      .
 260 PRINT Z$(N);
1000 DIM A$(100,10)
1010 DIM B$(100,10)
      .
      .
      .
      .
1250 DIM Z$(100,2)
2000 INPUT A$(N)
2010 INPUT B$(N)
      .
      .
      .
      .
2250 INPUT Z$(N)
```

Second, use subroutines with "pointers", such as

```
10 PRINT A$(N);
15 IF C=1 THEN RETURN
20 PRINT B$(N);
25 IF C=2 THEN RETURN
```

These pointer-controlled subroutines may also apply to DIM statements and certainly apply to INPUT statements. This approach allows considerable flexibility in input/output. The demonstration routine below prints the first ten lines of arrays A, B, and C, and also ten lines of only A string (ie A$(1) through A$(10)).

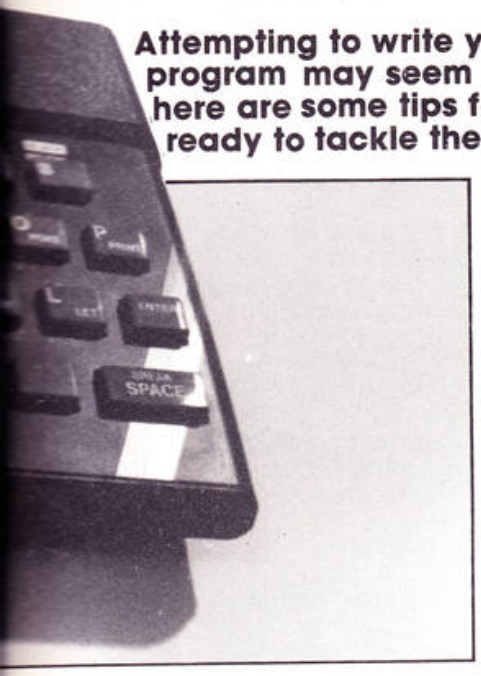
```
3000   PRINT "THIS PRINTS THREE
       COLUMNS"
3010   FOR N=1 TO 10
3020   LET I=1
3030   LET C=3
3040   GOSUB (I*10)
3050   PRINT
3060   NEXT N
3090   PRINT "THIS PRINTS ONE
       COLUMN"
3100   FOR N=1 TO 10
3110   LET C=1
3090   GOSUB (I*10)
3100   PRINT
3110   NEXT N
```

Consideration must be given to the fact that as the program listing increases in size, the

BASE PROGRAMS

BY MERVIN J CAGLE

Attempting to write your own data base program may seem like a daunting task, but here are some tips for those of you who are ready to tackle the job.



maximum possible size of the data base decreases (the reverse is also true). In cases where memory is at a premium, I would opt for making maximum use of the immediate or command mode. An entry like

```
LET E$(4)= "1st Class"
```

will facilitate adding data and an entry such as

```
LET E$(4)= ""
```

will delete data (ie replace the contents of string array defined as E$(4) with the empty string). A technique such as this can reduce listing size and free up valuable memory space. DIM statements can also be entered in the direct or command mode. Of course, user needs and memory availability dictate these choices.

Easy as A,B,C . . .

A routine to alphabetize may not be practical when it is easier to drop data down one line and enter new data on the available line (i.e. don't let the computer do something you can do easier). If you want to enter Johnson in your data base, you simply move all data after James down one line.

24 Jackson	24 Jackson
25 James	25 James
26 Jones	26 Johnson
	27 Jones

Normally, it is better to keep the program as simple as possible with easy to define lines and columns; however, a slightly more complex single string format may have its attractions. Earlier, we defined one hundred lines (or rows) consisting of columns A$ through Z$. An alternative is to use only one string variable such as A$ and segment the string into columns. In this example,

```
10 DIM A$(500,10)
20 FOR N=1 TO 100
30 INPUT A$(N)
40 INPUT A$(N*100)
50 INPUT A$(N*200)
60 INPUT A$(N*300)
70 INPUT A$(N*400)
80 PRINT A$(N); A$(N+100); etc.
```

A$(N+100) is column 2, A$(N+200) is column 3, and so on. Another possiblity is to use a legend. Just as legends reduce the size of wall mounted status boards, we can use a legend to reduce memory requirements in a computer program. Note this approach

```
A$ (1)= "ALAN"   X$ (1)= "CLERK"   Y$ (1)= "2"
A$ (2)= "ATKINS" X$ (2)= "DRIVER"  Y$ (2)= "2"
A$ (3)= "BARNS"  X$ (3)= "SALES"   Y$ (3)= "1"
```

If you wanted to print Mr. Barns' occupation, the following statement would accomplish the purpose.

```
150 PRINT A$ (3),
160 PRINT X$ (VAL Y$ (3) )
```

As mentioned previously, alphabetical sorting is not necessary when names are always entered in alphabetical order. This sub-routine opens up a new space at line L in a data base consisting of 120 lines (careful: line 119 moves to 120 — data originally in line 120 is lost)

```
4000 LET M=120
4010 LET A$(M)=A$(M-1)
4020 LET B$(M)=B$(M-1)
      .
      .
      .
4200 LET M=M-1
4210 IF M=L THEN RETURN
4220 GOTO 4010
```

Actually line L and line L+1 become identical; however, line L is to be written over with new data.

This routine can be used to delete a line of data at L.

```
4300 FOR M=L TO 120
4310 LET A$(M)=A$(M+1)
4320 LET B$(M)=B$(M+1)
      .
      .
      .
4350 NEXT M
```

Getting friendly

In order to have a user friendly program, the use of INKEY$ has a lot of advantage over input statements. One good application is in menu routines.

```
MENU

1. ADD
2. DELETE
3. REVIEW

PLEASE SELECT BY PRESSING NUMBER
```

We could choose to put in an INPUT statement which requires two keystrokes during program execution (a number and enter). On the other hand, statements using INKEY$ avoid the need for pressing the enter key, for example:

```
5045 SLOW
5050 IF INKEY$="" GOTO 5050
5051 IF INKEY$="1" GOTO 6000
5052 IF INKEY$="2" GOTO 6100
5053 IF INKEY$="3" GOTO 6200
```

Line 5050 is a one line loop which continues until a key is pressed. Lines similar to 5050 are found in many game-type programs. My philosophy is to use INKEY$ in all cases where a single keystroke branches you to the routine you select. INPUT statements are required for entering data unless the data is capable of being entered with a single keystroke. (Grade A, B, C, etc. would be an example of data entry using a single keystroke). INKEY$ is also handy in being able to "press any key to continue".

If a menu scheme is written into the program, it is important to remember that the screen should display as many selections as possible. For instance, ten screen displays of one line yes/no questions creates user frustration. Clearly, the number of keystrokes needed to get to the desired display must be minimized.

Look through any window

Another routine often found in video game programs can be used to create a "moving window" display.

```
100 IF INKEY$="" THEN GOTO 100
110 IF INKEY$="1" THEN LET L=L+1
120 IF INKEY$="6" THEN LET L=L-1
130 IF INKEY$="5" THEN LET C=C+1
140 IF INKEY$="8" THEN LET C=C-1
```

A touch of a key moves the display up, down, right, or left. It is possible to imitate some of the qualities of commercial software programs which allow users to quickly shift ranks and files of data at the touch of a key. Often column 1 is displayed continuously while others are shifted. The relatively slow speed of the ZX81 will dictate use of FAST mode; however, the slowness is only a minor inconvenience compared with the ability to move data in the direction you desire. Additional INKEY$ statements which move data in larger increments such as ten lines per keystroke (IF INKEY$="U" THEN LET L=L+10) may provide relief to the speed problem.

Some frustration often occurs when printing is interrupted with report code 5 (which indicates screen full) or when the printout has a diagonal alignment instead of a smooth vertical alignment. As easy way to solve the full screen problem is to ensure that only 22 lines are displayed at one time. The diagonal alignment problem can be remedied by ensuring displayed column widths equal 32 characters. That is, if A$, B$, C$, and D$ are displayed the LEN A$+LEN B$+LEN C$+LEN D$=32. If undisplayed columns are to be displayed, they must have the same column widths as those replaced. If A$, C$, D$, and E$ are to be displayed, then LEN E$ must equal LEN B$ which was the column replaced.

Good relations

A data base has little use if it cannot answer your questions. That is — the data base must be "relational". The easiest way to relational I/O is to be able to compare an inputed item of data to data in all lines of a particular column. Better yet — inputed data (plural) to be compared to applicable columns (plural) in all lines. In addition, a partial field search is also a nice feature. Finding everyone's name which begins with the letter "R" is very handy. A successful relational routine should have a statement similar to this for every string variable defining a column of data.

```
300 IF A$(Z,P TO Q)=A$(N,P TO Q)
THEN RETURN
305 GOTO 6398
```

For queries that look for an absence, such as a command to "print everything", statements like

```
310 IF A$(Z,P TO Q)=A$(N,P TO Q)
THEN RETURN
315 GOTO 6398
```

allow for this. If line 310 is affirmed (ie A$(Z,P TO Q) is not equal to A$(N,P TO Q), then the search continues on the same line to the next column to be compared. If line 310 is not affirmed, then the line is ignored for printing and the program proceeds to line 315 which will GOTO 6398 which is NEXT N (next line). Normally, A$(Z,P TO Q) will be entered as the empty string "", and when A$(N,P TO Q) equals other than a blank space, it will be affirmed and the RETURN to a PRINT statement executed. Thus, everything (non-blank spaces) will be printed (eg all strings with contents from A$(1,P TO Q) to A$(120,P TO Q), assuming a data base of 120 lines. Space precludes a full treatment of this process; however, most serious enthusiasts should get the idea.

A single string array can be set up for an entire data base. If A$ is dimensioned as a large "spread sheet", such as DIM (120,60). In this case, approximately 50% of the memory in a 16k RAM pack is set aside for this array. The other 50% of the memory can be used for the program listing. Columns like this example can be defined: Character 1 to character 12 for names, character 13 to 18 is for job (i.e. clerk, sec., jan., etc.) character 19 to 24 is for skill level or pay grade. The remaining columns can be defined in a like manner. This routine prints the contents of line two

```
10 LET N=2
20 LET P=19
30 LET Q=24
40 PRINT A$ (N,P TO Q)
```

from character 19 to character 24 which is the skill level/pay grade. The advantage of a single string array is in the ease of writing print routines and moving window routines. The disadvantage is that you cannot trim your array if your listing size begins running you out of memory.

If you are interested in writing a "home brew" data base program, you will be wanting to try some of these concepts. You will discover some techniques of your own, particularly if you have a penchant for assembly language. If you have some technique or insight to contribute, I am sure this magazine would welcome additional commentary on this subject. Some final words of advice — get used to typing GOTO instead of RUN, keep plenty of cassette tape backup, and save programs as soon as possible after changing either the program listing or the data base.

ALIEN ATTACK

Here's one for all you trigger-happy alien-zappers.

BY PAUL HAYWARD

You are Earth-bound but, luckily, provided with weapons, when a mother ship full of strange creatures and small craft approaches the planet. Now's your chance to save the world by stopping them from landing.

In this version of the arcade game you aim with the 'Z' and 'X' keys and fire with the Space key. Instead of bunkers there is an 'atmosphere' which will provide you with a limited amount of protection, and rapid random firing is prevented since continual firing will abort previously fired missiles. For each alien or ship destroyed you score 10 points, and you gain another 100 points by destroying the mother ship.

Variables

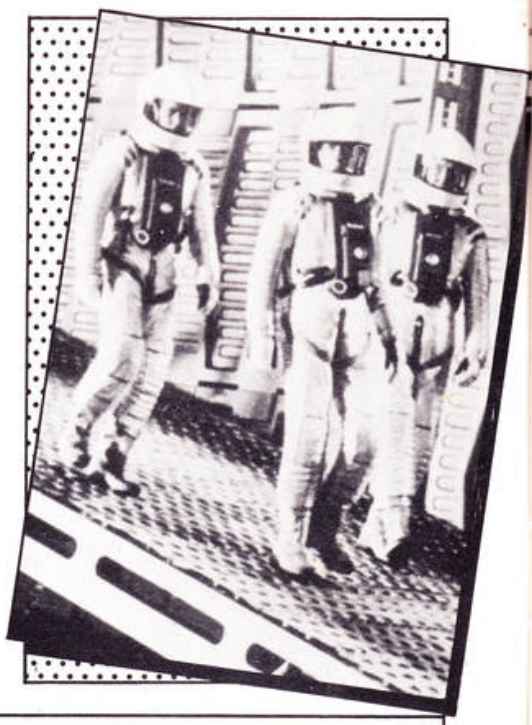
H	High score
S	Score
X	Vertical base coordinate
Y	Horizontal base coordinate
A	Vertical coordinate of first row of aliens
B	Horizontal coordinate of alien rows
L	Counter (starts at 0; when 9 is reached aliens move down vertically one row)
Z	Either 1 or −1, increments or decrements b
a( )	Switch (0 or 1) status is determined by whether there are aliens left in each row
Fi,Fa	Switches tell whether a missile has been fired
Fx,Fy	Coordinates of Earth missile
Ax,Ay	Coordinates of alien missile
F	Allotted for general use

Structure

10-60	UDG's, variables and set-up routines
100	PRINTS missile base
110	Scans keyboard for base movement
120	Checks if space bar has been used to fire
122	Erases previous missile
124	Sets new missile coordinates
125	Checks whether there is a missile on the screen
130,140	Updates missile
150-200	Prints aliens
210-320	Alien fires, various checks
400	Update alien's horizontal coordinates
500	Loops back to 100

Subroutines

1000	Earth missile has hit something, determines what and acts accordingly
2000	Alien missile has hit something, determines what and acts accordingly
5000	Die routine, new game option
6000	Instructions
8000	Set up variables, string variables, 8090 draws atmosphere
9000	UDGs, data for music



```
   1 REM **********************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         **********************
  10 REM Alien Attack
  20 GO SUB 6000: REM Instructio
ns
  30 LET HI=0: POKE 23658,8: REM
 set caps lock
  40 GO SUB 9000: REM UDG's
  50 LET y=INT (RND*28): LET X=2
0: LET s=0: LET LIVES=3
  60 GO SUB 8000: REM Set-up
 100 PRINT AT x,y;" A "
 110 LET y=y+(INKEY$="X" AND y<2
9)-(INKEY$="Z" AND y>0)
 120 IF INKEY$<>" " THEN  GO TO
125
 122 IF fi THEN  PRINT AT fx,fy;
" "
 124 LET fi=1: LET fx=x-1: LET f
y=y+1: LET Y1=y
 125 IF NOT fi THEN  GO TO 150
 130 PRINT AT fx,fy;" ": LET fx=
fx-1: IF SCREEN$ (fx,fy)="" THEN
  GO TO 1000
 140 PRINT AT fx,fy; INK 6;"E":
IF NOT fx THEN  PRINT AT fx,fy;"
 ": LET fi=0
 150 IF a(1) THEN  PRINT AT a,b;
 INK 2;a$
 160 IF a(2) THEN  PRINT AT a+2,
b; INK 1;b$
 170 IF a(3) THEN  PRINT AT a+4,
b;c$
 200 PRINT AT 0,0;Z$(X1 TO X1+31
): LET X1=X1+1: IF X1=74 THEN  L
```

```
ET X1=1
 210 IF fa THEN  GO TO 300
 220 LET ay=INT (RND*3)+(y-1)
 225 IF SCREEN$ (a+4,ay)="" THEN
  LET ax=a+5: GO TO 260
 230 IF SCREEN$ (a+2,ay)="" THEN
  LET ax=a+3: GO TO 260
 240 IF SCREEN$ (a,ay)="" THEN
LET ax=a+1: GO TO 260
 250 GO TO 400
 260 IF ATTR (AX-1,AY)=38 THEN
GO TO 400
 270 PRINT AT AX,AY;"E": LET FA=
1
 300 IF ax>=20 THEN  PRINT AT ax
,ay;" ": LET fa=0: GO TO 400
 310 PRINT AT ax,ay;" ": LET ax=
ax+1: IF SCREEN$ (ax,ay)="" THEN
  GO TO 2000
 320 PRINT AT ax,ay;"E"
 400 LET b=b+z
 430 IF b>0 AND b<16 THEN  GO TO
 100
 440 LET z=-z: LET l=l+.3: IF l<
.9 THEN  GO TO 100
 450 IF a(1) THEN  PRINT AT a,b;
e$
 460 IF a(2) THEN  PRINT AT a+2,
b;e$
 470 IF a(3) THEN  PRINT AT a+4,
b;e$
 480 LET l=0
 499 LET a=a+1: IF a=(15 AND a(3
)) OR A=(16 AND A(2)) OR A=(17 A
ND A(1)) THEN  GO TO 5000
 500 GO TO 100
1000 REM
1010 PRINT AT fx,fy;"E": BEEP .5
,-30
```

```
1012 IF ax=fx AND ay=fy THEN  LE
T fa=0: GO TO 1080
1015 LET h=y1-b+1: IF Z=1 THEN
LET H=H+2
1016 IF H>17 THEN  LET H=16
1017 IF H<1 THEN  LET H=2
1018 IF NOT FX THEN  LET S=S+90:
 PRINT AT 0,fy-1;"EEE": LET x1=1
: GO TO 1070
1020 IF fx=a THEN  LET a$(h)=" "
: GO TO 1060
1040 IF fx=a+2 THEN  LET b$(h)="
 ": GO TO 1060
1050 IF fx=a+4 THEN  LET c$(h)="
 "
1060 IF a$=e$ THEN  LET a(1)=0
1065 IF b$=e$ THEN  LET a(2)=0
1068 IF c$=e$ THEN  LET a(3)=0
1070 LET S=S+10: PRINT AT 21,8;
PAPER 2; INK 6;S
1085 PRINT AT fx,fy;" "
1090 IF NOT (a(1)+a(2)+a(3)) THE
N  GO SUB 8000
1095 LET FI=0: GO TO 100
2000 PRINT AT ax,ay;"E": BEEP .5
,-40
2010 IF ax=fx AND ay=fy THEN  LE
T fi=0: GO TO 2060
2030 IF ax=20 THEN  LET lives=li
ves-1: PRINT AT 21,19; PAPER 2;
INK 6;lives: IF lives<1 THEN  GO
 TO 5000
2060 LET FA=0
2070 PRINT AT ax,ay;" "
2080 GO TO 100
5000 REM
5010 IF hi<=s THEN  LET hi=s
5050 RESTORE 9200: FOR F=1 TO 11
: READ LENGTH: READ PITCH: BEEP
LENGTH,PITCH: NEXT F
5070 PRINT AT 10,2; FLASH 1;"New
 Game?(y/n)"
5080 IF INKEY$="" THEN  GO TO 50
80
5085 IF INKEY$="Y" THEN  GO TO 5
0
5086 GO TO 9999
5099 STOP
6000 PRINT AT 6,8;"Alien Attack"
6010 PRINT '"  ""Z""=Left"''"  "
"X""= Right"
6020 PRINT ,,,,TAB 6; FLASH 1;"S
PACE BAR"; FLASH 0;" to fire!"
6080 RESTORE 9300
6085 FOR F=1 TO 16: READ LENGTH:
 READ PITCH: BEEP LENGTH,PITCH:
NEXT F
6090 PAUSE 60: BEEP .5,0
6099 RETURN
8000 REM
8005 BEEP .04,-12: BEEP .03,-13:
 BEEP .04,-9: BEEP .03,-12:  BEE
P .04,-6: BEEP .05,-4
8020 LET a=5: LET b=0: LET z=1:
LET l=0
8025 DIM a(3): FOR f=1 TO 3: LET
 a(f)=1: NEXT f
8026 LET m=0
8030 LET fi=0: LET fa=0: LET fx=
x-1: LET fy=0: LET AX=0
8050 LET A$=" B B B B B B B B ":
 LET B$=" C C C C C C C C "
8055 LET C$=" D D D D D D D D "
8070 LET e$="                "
8075 LET Z$=E$+E$+E$+"GH"+E$+E$+E$+
E$: LET X1=1
8080 BORDER 0: INK 0: PAPER 4: C
LS
8085 PRINT AT 21,2; PAPER 2; INK
 6;"SCORE:";S;TAB 13;"LIVES:";LI
VES;TAB 23;"HIGH:";HI
8090 PLOT 0,40: DRAW 254,0,-.4
8099 RETURN
9000 RESTORE 9000: FOR f=USR "A"
 TO USR "I"+7: READ a: POKE f,a:
 NEXT f
9010 RETURN
9090 DATA 24,24,24,60,60,60,126,
255
9110 DATA 24,60,90,126,24,36,66,
0
9120 DATA 0,56,84,56,124,170,170
,170
9130 DATA 0,16,56,84,124,170,130
,68
9140 DATA 0,0,24,60,60,24,0,0
9150 DATA 2,64,20,33,84,32,8,66
9160 DATA 1,7,28,62,115,65,32,0:
 DATA 128,224,56,124,206,130,4,0
9170 DATA 60,126,219,126,90,129,
66,36
9200 REM lose song
9210 DATA .6,-20,.6,-20,.2,-20,.
6,-20,.6,-17,.2,-18
9220 DATA .4,-18,.2,-20,.4,-20,.
2,-20,.6,-20
9300 REM Theme
9310 DATA .5,-12,.3,-10,.25,-8,.
1,-7.2,.08,-5,.55,-4
9320 DATA .12,-5,.08,-6,.2,-3.8,
.3,-7
9325 DATA .15,-9,.25,-7.5,.3,-10
9330 DATA .12,-12,.3,-10.5,.4,-1
3
9340 DATA .5,-12,.3,-10,.25,-8,.
1,-7.2,.08,-5,.55,-4
9350 DATA .2,-4,.3,-2,.4,0.5,.2,
1.5,.2,0.5,.3,-1,.08,-2,.15,-2.5
,.08,-3,.4,-4
9990 REM A B C D E F GH I
9991 REM A B C D E F GH I
9999 STOP
```

BOOKFILE

BY R DICKSON

A really useful program for ZX81 users who deal with editing and processing text.

Bookfile stores up to 17 pages of 20 lines of information in book form, plus a text page, all of which can be edited and processed quickly for ZX81 users.

First, enter the program and then the following constants, variables, unlisted. Use keywords TO, INPUT, RETURN, LIST in statements 244, 250 and 398.

Constants	Variables
LET B=0	LET E=0
LET C=9	LET N=1
LET D=640	LET P=1
LET F=32	LET Q=20
LET H=240	LET R=14
LET I=18	LET X=0
LET J=1	LET A$=""
LET K=2	LET B$=""
LET O=28	LET D$=""
LET S=6	DIM E$ (K)
LET T=10	DIM C$ (J,D)
LET W=128	LET K$=">"
LET Y=20	

LET G$=" [illegible]

G$ is the graphics string, made up of graphic characters and blanks on the keys 0-8 and A-Z, so that the first character is a blank followed by inverse blank, graphics 1-8, blank, Graphic A, blank, blank, graphic D-H, eight blanks, graphic Q-T, two blanks, graphic W, blank, graphic Y, blank. That makes 37 characters and will enable all graphics characters to be drawn, in key order, without further trouble, in the program.

Getting started

GOTO 6 will start the program and the book will open with a minimum of one page in file and one page text. The number at the top right hand corner is the book length and can be set from 1 to 17 pages, depending on the size of book you want to start with.

In order to get used to the many operating facilities, it would be a help to type in at least two pages of operating instructions which can be recorded with an open book for subsequent reference.

When you have entered the program, constants and variables, save it by getting your recorder ready and pressing the Shift Q key which will automatically save the program and operating data, on a minimum book-size setting. This will take approximately 1 minute 45 seconds to save, as opposed to a full book of 17 pages which will take about 5 minutes 45 seconds.

Get exploring

You may wish now to explore the possibilities of Bookfile which is in two forms, text mode and edit mode. Text mode enters characters onto the text page plus spacing facilities as listed in the operating data. Full inverse and graphics mode facilities are available by pressing the shift 9 key, when the text cursor will change from > to < (inverse mode) to = (graphics mode) and back to > (text mode normal).

Edit mode (shift 1) changes

the cursor to −, when it now becomes possible to insert or change text rapidly using direction keys (shift 5-8) for rapid location of the cursor into the text. Insert any "mode" into text at any time (in Edit mode) and when finished, exit by pressing Edit (shift 1) again, back to text mode.

The text can be transferred into file by using the TO key (shift 4) or by filling the page by tripping the end + cursor, with an auto-file facility, with text.

In order to set the book to maximum, press AND (shift 2) and enter 17. This will preserve whatever is in file 1 and set the book length to maximum.

edit mode in case that ever appears to become hidden in the text.

TO will push text into file and advance a fresh page if the file was empty, or present that file for review or deletion if occupied.

Reset each new book before use as any file above 1 will then be cleared on renewal.

THEN clears the current line, then tabs back to the previous line adjustment point (✱).

Edit (inverse space) is 0

using = (graphics) cursor or

NEWLINE in < (inverse) mode.

Retain the — cursor at the end of the text position in edit mode, by first moving the cursor upwards with the shift 7 key.

variables, files and text.

The remedy is simply to re-load.

The final page will turn onto page 1 but can easily be retrieved using the FAST key (shift F).

If the BREAK key is accidentally pressed, do not worry — simply enter GOTO 6, in text mode, or GOTO 200 in edit mode, and the text will return intact.

Resetting the book will set the text page always to 1 — this cuts down save-time.

In graphic mode, non-graphic, single-token keys will print out the same along with the bottom row of keys.

Great subsids

A subsidiary mode, the file review mode, is obtained using FUNCTION, which scans through the file giving the character count of whatever is in file, plus page information. Input 0 to exit back to text mode.

OR scoops the current file with text page in text mode, a useful key, and it flashes the cursor in

If the end + marker is ever erased (very rare), reset this simply by varying the page number with shift 5 or 8.

The text mode writing speed is approximately 1.5 times faster than edit mode, so allow a little slower entry for edit mode entry and for < = mode entries too.

Avoid using the BREAK key all spacing is done using the NEWLINE key and never use RUN or CLEAR as these will clear the

And so

With a little familiarity and practice, the whole character availability of your ZX81 will now be at your disposal for creating any kind of text in book form. All data in file and on text can be printed using the LPRINT and LLIST keys, and page numbers will be central for file print-outs and bordered for the text page.

So now you know all about this great program, get typing and write your own book.

```
   6 CLS
   8 PRINT TAB C;"TEXT PAGE";TAB
 Q;N
   9 PRINT AT B,Y;P,AT Y+J,F-J;"
+"
  10 PRINT AT K,B;B$;A$;
  18 PRINT K$;AT 24-PEEK 16442,L
EN A$;
  20 LET A=CODE INKEY$
  22 IF A=8 THEN GOTO Y
  24 IF A>=V THEN GOTO A+A
  26 IF E=J THEN GOSUB V+X
  28 PRINT CHR$ A;
  29 LET A$=A$+CHR$ A
  30 IF LEN A$>=F THEN GOSUB F
  31 GOTO I
  32 LET B$=B$+A$
  34 LET L=LEN A$
  36 LET A$=""
  38 LET M=LEN B$
  40 IF M<D THEN RETURN
  80 LET D$=B$+A$
  82 LET M=LEN D$
  84 LET B$=C$(P, TO INT (C(P)/F
)+F)
  86 LET A$=C$(P,LEN B$+J TO C(P
))
  88 IF A=V+V THEN GOTO V-I
  90 LET C(P)=M-(J AND M=D)
  92 LET C$(P)=D$
  94 LET D$=""
  96 LET G=M-LEN B$-LEN A$
  98 DIM E$(G+(J AND M<D-J) OR G
<J)
 100 PRINT AT K,B;B$;A$;E$
 102 DIM E$(K)
 104 GOTO T+(M-Y AND A<PEEK T)
 112 LET A=A+W
 113 IF A>=0 AND X=J THEN LET A=
CODE G$(A-Y-S)
 114 IF A=8 THEN GOTO I+(D AND L
EN B$>M)
 115 RETURN
 118 LET X=J AND X=8 AND E=J
 120 LET E=J AND (E=8 OR X=J)
 122 LET K$=CHR$ (I+X+E+(K+K AND
 E=8 AND LEN B$>M))
 124 IF LEN B$>M THEN RETURN
 126 GOTO I
 128 IF LEN A$<=0 THEN GOTO W+S
 130 LET A$=A$( TO 0)
 132 GOTO V+0
 134 IF LEN B$=8 THEN GOTO H
 136 LET A$=B$(LEN B$-F+J TO LEN
 B$-(F-0 OR A=W-C))
 138 LET B$=B$( TO LEN B$-F)
 140 IF A=W-C THEN GOTO H
 142 DIM E$(F+J-(0+K AND LEN B$)
=D-F))
 144 GOTO W-0
 146 DIM E$(F)
 148 IF LEN A$<0 OR LEN B$>=D-F
THEN GOTO W+0
 150 GOSUB F
 152 LET A$=E$( TO 0)
 154 LET B$=B$+E$( TO F-L)
 156 IF LEN A$<0 THEN LET A$=A$+
E$( TO 0-LEN A$)
 158 PRINT " "
 160 GOTO V-T
 168 GOSUB F+K
 170 LET B$=B$ AND A=M-I
 172 DIM E$(L+(M AND A>V+V)+(J A
ND M+L<D-J))
 174 GOTO W-0
 200 LET K$=">"
 202 LET E=8
 204 LET A$=B$(M+J-L TO M)
 206 LET B$=B$( TO M-L)
 208 GOTO T-(K AND A=8)
 226 LET 0=LEN A$
 227 GOTO W+(I AND A=V+J)
 228 LET P=P-J OR P=J
 230 IF A<>V+K THEN LET P=P+J OR
 P=N
 231 GOTO C
 232 GOTO W-T
 234 GOTO D-V
 236 LET A=W AND E>X
 237 GOTO 0
 238 IF LEN A$=8 THEN GOTO W+S
 239 LET A$=A$( TO LEN A$-J)
 240 PRINT " "
 241 GOTO T
 242 CLS
 244 PRINT "  INPUT PAGE (1-";N;
")    RETURN /0"
 246 INPUT G
 248 IF G<J OR G>N THEN GOTO S
 250 PRINT AT B,K;" LIST PAGE ";
G;", ";C(G);E$,,,C$(G)
 252 GOTO H+S
 339 PRINT B$(G+J) AND G<M
 340 IF A=V THEN LET G=G-(F AND
G)=F)
 341 IF A=V+J THEN LET G=G+(F AN
D G<=M-F)+(M-G AND M-G<F)
 342 IF A>V+J THEN LET G=G-(J AN
D G>8)
 343 PRINT AT INT (G/F)+K,G-INT
(G/F)*F;K$;B$(G+K) AND G<M
 344 GOTO D-T
 345 GOTO D+K
 348 GOSUB W-T
 349 GOTO H+V-C
 351 GOTO Y*T
 354 LET A=8
 355 GOTO D
 357 LET B$(G+J)=""
 358 GOTO H+V-T
 363 GOTO PI*V
 384 SAVE "■"
 390 CLS
 398 PRINT " HOW MANY PAGES? (1-
17)";;;" TO PRESERVE FILE";"S" A
ND N>J;" ";N," INPUT 0"
 400 INPUT G
 402 IF G>I-J THEN GOTO Y*Y
 404 IF G=8 THEN GOTO S
 406 PRINT AT J+K,S;G
 408 LET P=J
 410 LET N=G
 412 LET D$=C$(J, TO C(J))
 414 DIM C$(N,D)
 416 LET C$(J)=D$
 418 DIM C(N)
 420 LET C(J)=LEN D$
 422 LET D$=""
 424 GOTO S
 432 GOTO D+C
 434 GOTO V-F
 436 GOTO PEEK Y+K
 440 LET 0=R
 441 GOTO W+(I AND A=M-Y)
 442 LET R=LEN A$
 443 GOTO H
 444 GOTO W+(PEEK W AND LEN A$>8
)
 446 LET A=J
 448 GOTO V-F
 450 LPRINT ;P,,,C$(P)
 451 GOTO H-T
 452 LPRINT P,,,,B$;A$
 453 GOTO T
 454 GOTO 0+S
 458 LET P=N OR A=CODE " SLOW "
 460 GOTO C
 528 LET E=8
 530 GOSUB F
 532 LET B$=B$+E$
 534 LET G=M
 536 LET K$="-"
 538 PRINT K$
 630 LET A=CODE INKEY$
 632 IF A=8 THEN GOTO D-T
 634 PRINT AT INT (G/F)+K,G-INT
(G/F)*F;
 636 IF A>=V THEN GOTO A+A+A
 638 IF E=J THEN GOSUB V+X
 640 LET B$(G+J)=CHR$ A
 642 PRINT B$(G+J) AND G<M;K$
 644 LET G=G+(J AND G<M)
 645 GOTO D-T
 649 LET A=S+S+(W AND E=J)
 650 GOTO 0+(D-0 AND LEN B$>M)
 651 PRINT B$(G+J)
 700 GOTO H+V-C
```

FRENCH LANGUAGE TEST

Test your French vocabulary with this program for the 16K ZX81

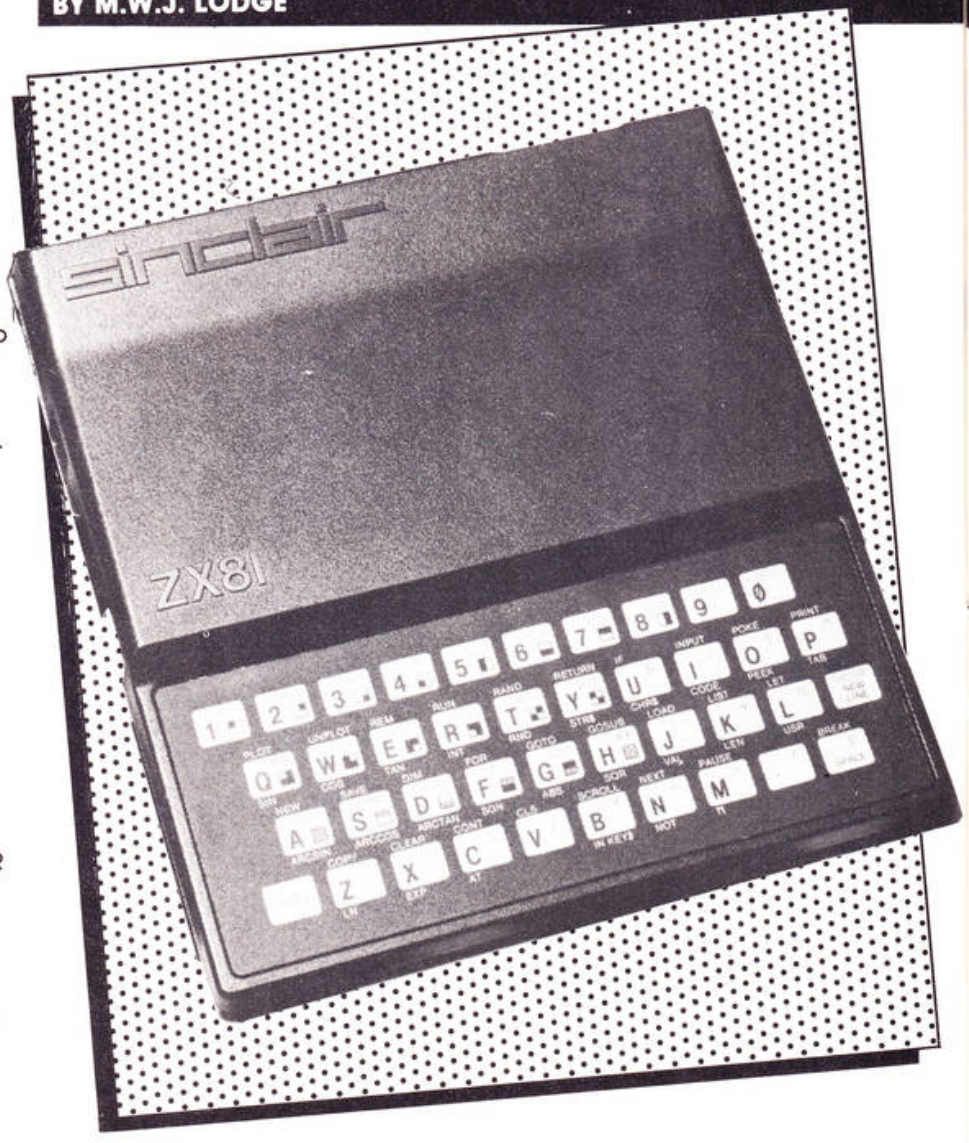
BY M.W.J. LODGE

This program was originally written to test French vocabulary, but it could be converted for other languages or even other subjects. The words are held in two arrays and, for a 16K ZX81, this allows a total capacity of 300 French words and their English equivalents, both with a maximum of 20 characters. These parameters could be varied to suit a student's individual needs and, because of program loading time, the length of the arrays (T) should be considered before adding any words. For example a student following a course of study might prefer a different version of the program for each chapter.

Pairing up

The program has the options to add new words, in pairs (French and English), to the array limit. To similarly change pairs of words though the word number, obtained in use from the following option, must be known. And finally there is an option to test either French or English vocabulary. When new words are added or existing words corrected the program must be SAVED. And after RUNning the program for the first time subsequent running must commence with the instruction GOTO 100, otherwise the words will be clared from the arrays.

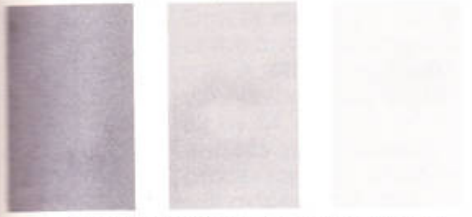
Both French and English words should be entered on the format "word (added clarification)", eg "L'ENCRE (FEM)", "TES (PLURAL)" and "YOUR (PLURAL)". The clarification is an option which will not always be necessary and the 20 character space limitation does not encourage its use. But when used it must be preceded by an open bracket, which also



identifies the end of the word, though the closing bracket is optional. The use of accents is difficult but essential for foreign languages so some improvisation is suggested, eg grave "graphics 2" before the letter and acute "graphics 1" after the letter. This might seem laborious but when being tested only the entry of alphabetic characters is required.

Passing the test

The test options give the student the choice of being examined in either French or English. The questions will be asked in random sequence and the reply should be entered without any clarification, ie the information



that might have been included within brackets. The computer will then extract and close up the alphabetic characters from both the answer and reply up to any open bracket that may be present. For example the reply "L ENCRE" becomes LENCRE, and the computer held answer "L'ENCRE (FEM)" also becomes LENCRE. Both are then padded with spaces to a common length and a comparison made. Therefore a reply could be entered without any spaces or even with excessive spaces.

After the comparison the correct answer from the array will be displayed, along with the actual reply as entered and a message to advise whether right or wrong. And finally the cululative score, a total and percentage, together with the option to continue.

This program should provide a valuable study aid for students of any age and could easily be converted to run on a Spectrum. The advantages of the latter would be having the capacity for an even larger array as well as the faster loading capability.

Lines

0-99	Variable data which must be retained.
100-199	Provides the initial menu screen and determines the processing required.
200-299	Allows words to be added to both arrays to the specified limit.
400-499	Tests French by displaying English words in random sequence.
500-599	Tests English by displaying French words in random sequence.
600-699	Allows existing words to be corrected.

GOSUB

1000	Generates a random number between 1 and the highest number of words in the array, but excludes consecutive duplicates. Therefore there must be at least two words in the array or this will loop.
1020	Keeps the score.
1030	Gives the option to continue or stop.
1040	Sets the variables required at run time.
1050	Accepts reply to question.
1060	Compresses both the reply and answer and pads out with spaces before making a comparison. This is a relatively slow process.

Variables

DIM F$	French array
DIM E$	English array
DIM A$	Input word array
DIM B$	Input working array
D$	Work field
X$	General alphabetic input
C	Control of words already added to both arrays
X	General numeric variable
A	Total attempts at test
B	Total correct answers to test
D	Duplicate control of random numbers
V	Used in GOSUB 1010
N	Used in GOSUB 1010
T	Controls maximum number of words specified for arrays.

```
  5 LET T=10
 10 DIM F$(T,20)
 20 DIM E$(T,20)
 30 LET C=0
100 CLS
105 SLOW
110 PRINT
115 PRINT "1 = ADD MORE WORDS"
120 PRINT "2 = CORRECT A WORD"
125 PRINT "3 = FRENCH TEST"
130 PRINT "4 = ENGLISH TEST"
135 PRINT "9 = END"
137 PRINT
138 PRINT "ENTER 1 2 3 4 OR 9"
140 INPUT X
145 IF X=1 THEN GOTO 200
150 IF X=2 THEN GOTO 600
155 IF X=3 THEN GOTO 400
160 IF X=4 THEN GOTO 500
165 IF X=9 THEN GOTO 175
170 GOTO 100
175 STOP
200 LET C=C+1
```

```
205 CLS
206 IF C<T+1 THEN GOTO 210
207 PRINT "MAXIMUM WORDS ENTERE
D"
208 PRINT "TYPE ""GOTO 100"" TO
RE START"
209 STOP
210 PRINT "COUNT IS ";C
215 PRINT
220 PRINT "ENTER FRENCH WORD"
225 INPUT F$(C)
230 PRINT " ";F$(C)
235 PRINT
240 PRINT "ENTER ENGLISH WORD"
245 INPUT E$(C)
250 PRINT " ";E$(C)
255 PRINT
260 PRINT "IF O.K. ENTER Y ELSE
N"
265 INPUT X$
270 IF X$<>"Y" THEN GOTO 205
275 GOSUB 1030
280 IF X$="Y" THEN GOTO 200
285 GOTO 100
```

```
 400 GOSUB 1040
 402 CLS
 404 PRINT "FRENCH WORDS"
 406 PRINT "*************"
 410 GOSUB 1000
 412 PRINT "ENGLISH WORD  ";"NO.
  ";X
 414 PRINT
 416 PRINT "  ";E$(X)
 418 PRINT
 420 PRINT "WHAT IS THE FRENCH W
ORD"
 422 GOSUB 1050
 423 FAST
 424 GOSUB 1060
 425 LET D$=B$
 426 LET A$=F$(X)
 427 GOSUB 1060
 429 SLOW
 430 IF D$=B$ THEN GOTO 438
 431 PRINT "ERROR"
 432 PRINT " THE CORRECT ANSWER
IS"
 434 PRINT "   ";F$(X)
 436 GOTO 441
 438 PRINT "CORRECT - ";F$(X)
 440 LET B=B+1
 441 GOSUB 1020
 442 GOSUB 1030
 444 IF X$="Y" THEN GOTO 402
 446 GOTO 100
 500 GOSUB 1040
 502 CLS
 504 PRINT "ENGLISH WORDS"
 506 PRINT "**************"
 510 GOSUB 1000
 512 PRINT "FRENCH WORD  ";"NO.
 ";X
 514 PRINT
 516 PRINT "  ",F$(X)
 518 PRINT
 520 PRINT "WHAT IS THE ENGLISH
WORD"
 522 GOSUB 1050
 523 FAST
 524 GOSUB 1060
 525 LET D$=B$
 526 LET A$=E$(X)
 527 GOSUB 1060
 528 SLOW
 529 IF D$=B$ THEN GOTO 538
 530 PRINT "ERROR"
 532 PRINT " THE CORRECT ANSWER
IS"
 534 PRINT "   ";E$(X)
 536 GOTO 541
 538 PRINT "CORRECT - ";E$(X)
 540 LET B=B+1
 541 GOSUB 1020
 542 GOSUB 1030
 544 IF X$="Y" THEN GOTO 502
 546 GOTO 100
 600 GOSUB 1040
 605 CLS
 610 PRINT "ENTER WORD NO."
 613 PRINT
 615 INPUT A
 616 IF A<T+1 THEN GOTO 620
 617 PRINT "NUMBER NOT IN DICTIO
NARY"
 618 PRINT "TYPE ""GOTO 100"" TO
 RESTART."
 619 STOP
 620 PRINT A
 630 PRINT "ENTER FRENCH WORD"
 635 INPUT F$(A)
 640 PRINT "  ";F$(A)
 645 PRINT
 650 PRINT "ENTER ENGLISH WORD"
 655 INPUT E$(A)
 660 PRINT "  ";E$(A)
 665 GOSUB 1030
 670 IF X$="Y" THEN GOTO 605
 675 GOTO 100
```

```
1000 RAND
1004 LET X=INT (RND*C)+1
1006 IF X=D THEN GOTO 1000
1007 LET D=X
1008 PRINT
1009 RETURN
1010 LET V=1
1011 LET B$="
 "
1012 FOR N=1 TO 20
1013 IF A$(N)>="A" AND A$(N)<="Z
" THEN GOTO 1016
1014 IF A$(N)="(" THEN LET N=20
1015 GOTO 1018
1016 LET B$(V)=A$(N)
1017 LET V=V+1
1018 NEXT N
1019 RETURN
1020 LET A=A+1
1022 PRINT AT 16,10;"YOUR SCORE
IS"
1024 PRINT AT 17,10;B;"  OUT OF
 ";A
1025 LET X=(B*100)/A
1026 PRINT AT 18,10;INT (X);"  P
ERCENT"
1028 RETURN
1030 PRINT
1032 PRINT "MORE WORDS Y OR N"
1034 INPUT X$
1036 RETURN
1040 LET A=0
1042 LET B=0
1045 LET D=0
1046 DIM A$(20)
1047 DIM B$(20)
1048 DIM D$(20)
1049 RETURN
1050 INPUT A$
1052 PRINT " YOUR REPLY - ";A$
1054 PRINT
1056 RETURN
1060 LET V=1
1062 LET B$="
 "
1066 FOR N=1 TO 20
1068 IF A$(N)>="A" AND A$(N)<="Z
" THEN GOTO 1074
1070 IF A$(N)="(" THEN LET N=20
1072 GOTO 1078
1074 LET B$(V)=A$(N)
1076 LET V=V+1
1078 NEXT N
1082 RETURN
```

```
FRENCH WORDS
************

ENGLISH WORD  NO.  1

 POLICE

WHAT IS THE FRENCH WORD
 YOUR REPLY - GENDARME


ERROR
 THE CORRECT ANSWER IS
  LE FUZZ


          YOUR SCORE IS
          0  OUT OF  2
          0  PERCENT

MORE WORDS Y OR N
```

Print 64

Get 64 characters per line rather than the normal 32 on your Spectrum.

BY KAI WEBER

One of the many disadvantages of the Spectrum, as opposed to, say, Auntie Beeb's micro, is its very low character resolution — 32 by 24. Anyone who has endeavoured to write a word processor on a humble Spectrum will appreciate that it is very difficult to overcome the character problem, unless you happen to have software that can generate at the very least 64 characters per line. You may well be asking where you can get such software from — well, look no further! The machine code routine in this article will allow you, by a very slight alteration to your print statements, to output to the screen all the normal characters but at 64 per line instead of the normal 32. Both sizes of text can be mixed on the screen at the same time.

I was inspired to write the program after reading Mike Lord's article 'Spectrum Streams' (ZX Computing, August/September '83) in which he stated that the values in the system variable STRMS were in fact pointers pointing to the Channel Information area of RAM. So I thought to myself, why not make stream four point to a new printing sub-routine instead of the usual error routine by poking locations 23568/9 (see 'Spectrum Streams').

Getting in print

Fig. 1 is a disassembly of the 48K version of Print 64. Basically, what the routine does is to first change the Spectrum character set by altering location 23606/7 CHAR$ into something which is a little more suited to 64 characters per line ie each character occupies only the first four bits. Next, it will perform one of two operations. If the current print position (in 64 character mode) is an even number or zero it will simply print the character in the A register at the position. However, if the current x-axis print position is an odd number the character in the A register is first rotated right four times before being printed OVER 1 at the current position (OVER 1 is used because it means the character does not instantly delete the previously printed character). Finally, the character set is made to point back to the data held in ROM before returning. Sounds complicated? It is!

The machine code and character data occupy just over 1K of memory directly beneath the UDG's — 1215 bytes to be exact. To use the routine first type in listing one. You should then proceed to input all the numbers in Fig. 2a if you own a 16K or 2b if you own a 48K. The computer automatically compiles a checksum and when you have finished typing will inform you of any errors. If you do make any errors, I am afraid you will just have to enter all the numbers again. When everything is correct, the computer will inform you in 64 character mode and then save the code to tape. If you experience difficulty reading the text at first try altering the tuning of your TV set very slightly.

Using the routine

The only alterations needed to a simple print statement are:

1. The addition of '#4;' just before the text to be printed.
2. The addition of the seemingly useless squiggle (located under the 'A' key) just before the text. (Don't worry! This character does not actually get printed.)

So therefore, PRINT "text" would become PRINT #4;" (squiggle) text"

What it cannot do

For complicated reasons, the program cannot handle keywords or UDGs. It simply subtracts 128 from them to turn them into ordinary characters. It is also incapable of dealing with control characters, so therefore any FLASH, OVER, AT, TAB, INK etc, statements should be placed before the # character, ie

```
PRINT INK 1; AT
12,12; #4;"(squiggle)
text"
```

N.B: The numbers after AT and TAB are still expressed as values between 0 & 31 — the first character to be printed always appears on the left hand side of the current 32 character mode print position.

Unfortunately, for some reason unknown to me, the routine does not like the CHR$ statement. It simply refuses to print anything if you type, say,

PRINT #4;"(squiggle)";CHR$ 65 which should display an 'A'.

Get scrolling

If the screen fills up and the computer prompts 'scroll?', you will probably notice that the message is made up of small characters, but with a space between each. If you stop the computer scrolling, all output appears like the message. To restore normality, type PRINT #4;""

I said earlier that you needed a squiggle character at the beginning of the text but this is not necessary if the previous print statement was terminated by a semi-colon and you wish the next piece of text to buffer onto the previous. For example, suppose you wanted to print out the character set from space to copyright symbol, you would use a program something like this:

```
10 PRINT #4;"(squiggle)";: REM
the first print statement should
always contain a squiggle.
20 FOR F=32 TO 127
30 LET A$=CHR$ F: PRINT #4;A$;
40 NEXT F
```

The only reason for putting the character to be printed into the variable A$ is that, as mentioned previously, the routine does not like the CHR$ statement. Generally speaking, upper case output tends to look neater than lower case, which, in my opinion, often looks straggly.

When using the routine in your own programs, if you do not already have the machine code on board, you should place this line at the very beginning:

```
CLEAR USR "A"-1215: LOAD
""CODE : LOAD ""CODE
```

You should then save both blocks of code directly after your program. To save the code, first load it in using the above commands and then simply copy both lines with 'SAVE' in them from the BASIC program in listing 1.

Text editor

Listing two is a very simple text editor which sorts out text so that no word overlaps from one line to another. The program assumes that you have the machine code on board. When run, it will ask you to input some text which should be more than 64 characters in length to see what the routine does. Once entered, the program will display the edited text at 64 characters per line. The only way to get it onto printer paper is to use the COPY command —LPRINT #4 does not work properly. Listing two is not meant to be a full-blooded word processor, just an example of how the routine can be used — a word processor may well feature in another article sometime in the future. As can be seen from this article the potential applications of using different streams are very powerful.

Print 64 has very obvious advantages for people wishing to use their Spectrums for serious uses rather than ridding the complete and utter universe of Klingons or similar alien menace. Your only problem now for word processing on a Spectrum is getting a decent keyboard, a decent printer, a printer interface and last but not least the actual word processor software — watch this space!

```
  2 REM LISTING 1 - M.C. LOADER
  10 CLEAR USR "a"-1215
  20 PRINT "Please input numbers
 in Fig 1";CHR$ (97+(PEEK 23733=
255))
  25 LET checksum=0
  30 FOR f=USR "a"-958 TO USR "a
"-1
  40 INPUT b: POKE f,b
  50 LET checksum=checksum+b
  60 NEXT f
  70 LET check=52734
  80 IF PEEK 23733=255 THEN  LET
 check=54957
  90 IF check<>checksum THEN  PR
INT "ERROR IN THE DATA": STOP
 100 POKE 23582,161: POKE 23583,
34+(128 AND PEEK 23733=255)
 110 PRINT #4;"ALL MACHINE CODE
 ENTERED CORRECTLY.PLEASE INSERT
 A BLANK          CASSETTE AND PRE
SS RECORD.PRESS ANY KEY WHEN REA
DY TO SAVE."
 120 PAUSE 200
 130 SAVE "M.C."CODE USR "A"-958
,958: SAVE "STRMS DATA"CODE 2358
2,2
 140 PRINT #4;"VERIFY...": VERI
FY ""CODE : VERIFY ""CODE
```

```
   1 REM LISTING 2 - TEXT EDITOR
   6 CLS
  10 INPUT "PLEASE ENTER TEXT ";
t$
  20 IF LEN t$<64 THEN  GO TO 40
  30 PRINT #4;"";t$: GO TO 10
  40 LET c=0
  50 IF c=64 THEN  GO TO 100
  60 LET b$=t$(64-c)
  70 IF t$(65-c)=" " OR b$=",
R b$="." OR b$=":" OR b$="|" OR
b$="?" THEN  GO TO 110
  80 LET c=c+1
  90 GO TO 50
 100 LET c=0
 110 PRINT #4;"";t$( TO 64-c)
 120 LET t$=t$(65-c TO )
 130 IF t$(1)=" " THEN  LET t$=t
$(2 TO )
 140 GO TO 20
```

```
LIST
00010 ORG    65180
00020 CP     126
00030 RET    Z
00040 RES    7,A
00050 CALL   weber
00060 CALL   #09F4
00070 CALL   sincl
00080 LD     HL,65217
00090 LD     (65366),HL
00100 RET
00110 weber  LD HL,64154
00120 LD     (23606),HL
00130 RET
00140 sincl  LD HL,15360
00150 LD     (23606),HL
00160 RET
00170 ORG    65217
00180 CP     126
00190 JP     Z,strms
00200 RES    7,A
00210 CALL   weber
00220 CALL   sethl
00230 CALL   rotr
00240 CALL   dfcc
00250 PUSH   AF
00260 CALL   over1
00270 POP    AF
00280 PUSH   AF
00290 CALL   #09F4
00300 CALL   over
00310 POP    AF
00320 CALL   sethl
00330 CALL   rotl
00340 CALL   sincl
00350 strms  LD HL,65180
00360 LD     (65366),HL
00370 RET
00380 rotr   LD C,8
00390 PUSH   AF
00400 loop   LD B,4
00410 il     RR (HL)
00420 AND    A
00430 DJNZ   il
00440 INC    HL
00450 DEC    C
00460 JR     NZ,loop
00470 POP    AF
00480 RET
00490 rotl   LD C,8
00500 PUSH   AF
00510 loop2  LD B,4
00520 il2    RL (HL)
00530 AND    A
00540 DJNZ   il2
00550 INC    HL
00560 DEC    C
00570 JR     NZ,loop2
00580 POP    AF
00590 RET
00600 sethl  PUSH AF
00610 PUSH   DE
00620 LD     DE,8
00630 LD     B,A
00640 LD     HL,64154
00650 loop3  ADD HL,DE
00660 DJNZ   loop3
00670 POP    DE
00680 POP    AF
00690 RET
00700 dfcc   LD HL,23684
00710 DEC    (HL)
00720 PUSH   AF
00730 LD     BC,(23688)
00740 INC    C
00750 LD     (23688),BC
00760 LD     HL,(23684)
00770 POP    AF
00780 RET
00790 over1  LD A,(23697)
00800 LD     (23728),A
00810 LD     HL,23697
00820 SET    0,(HL)
00830 RET
00840 over   LD A,(23728)
00850 LD     (23697),A
00860 RET
```

Fig 1 - Disassembled listing

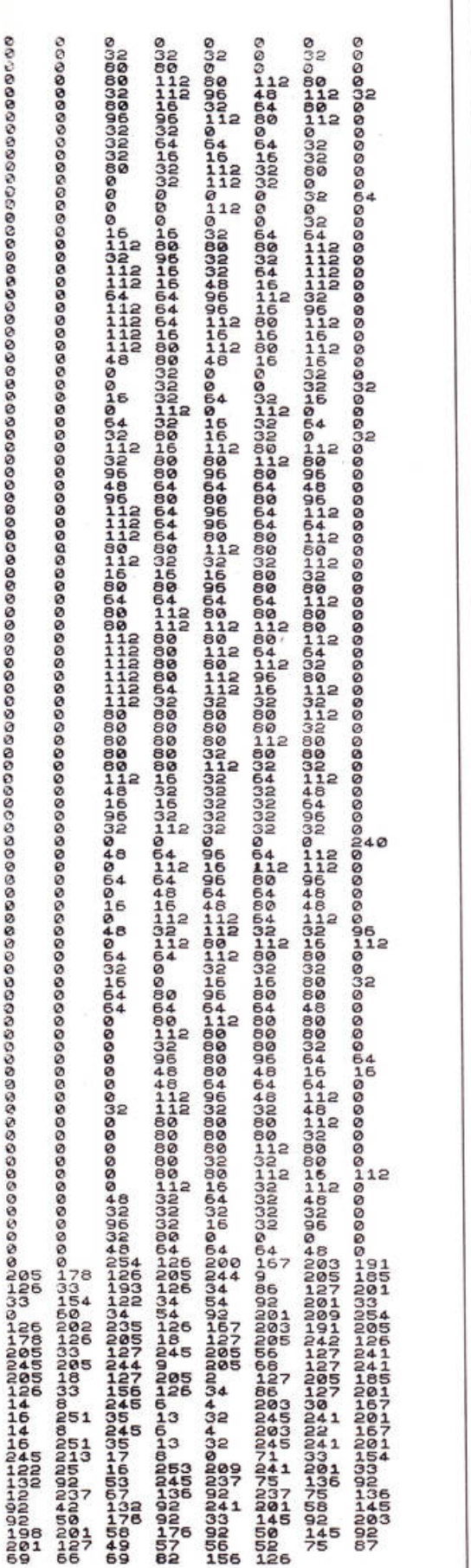


Fig 2a - 16K Machine Code

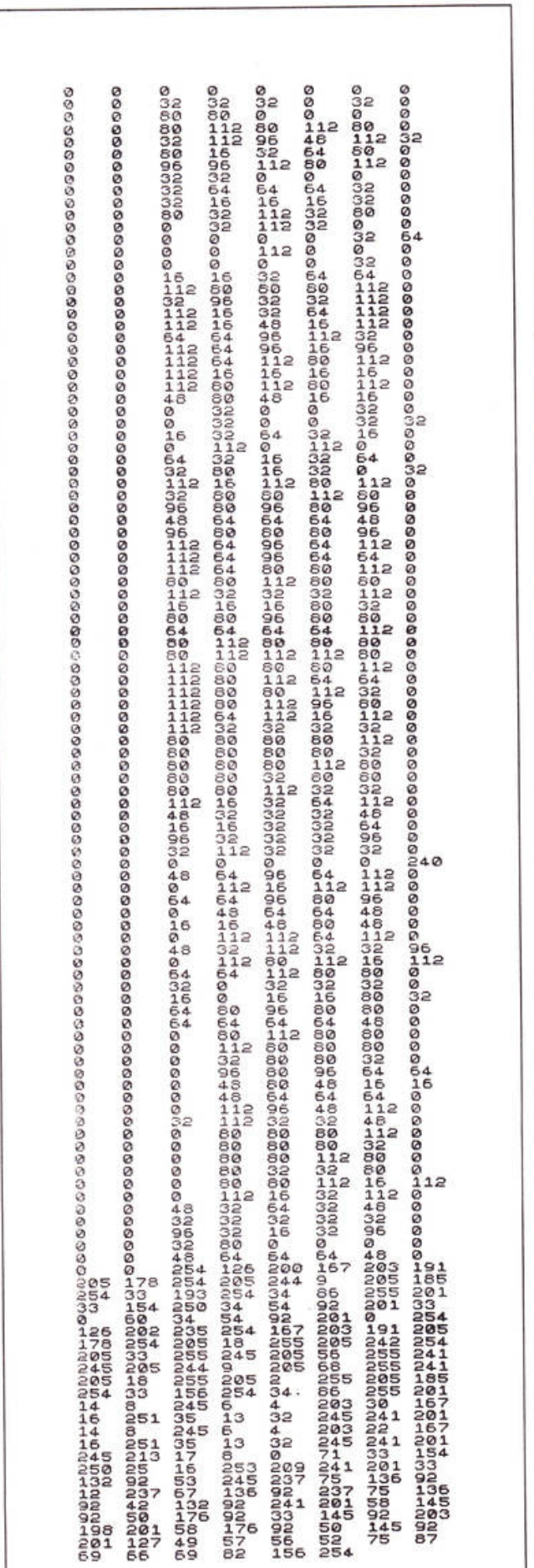


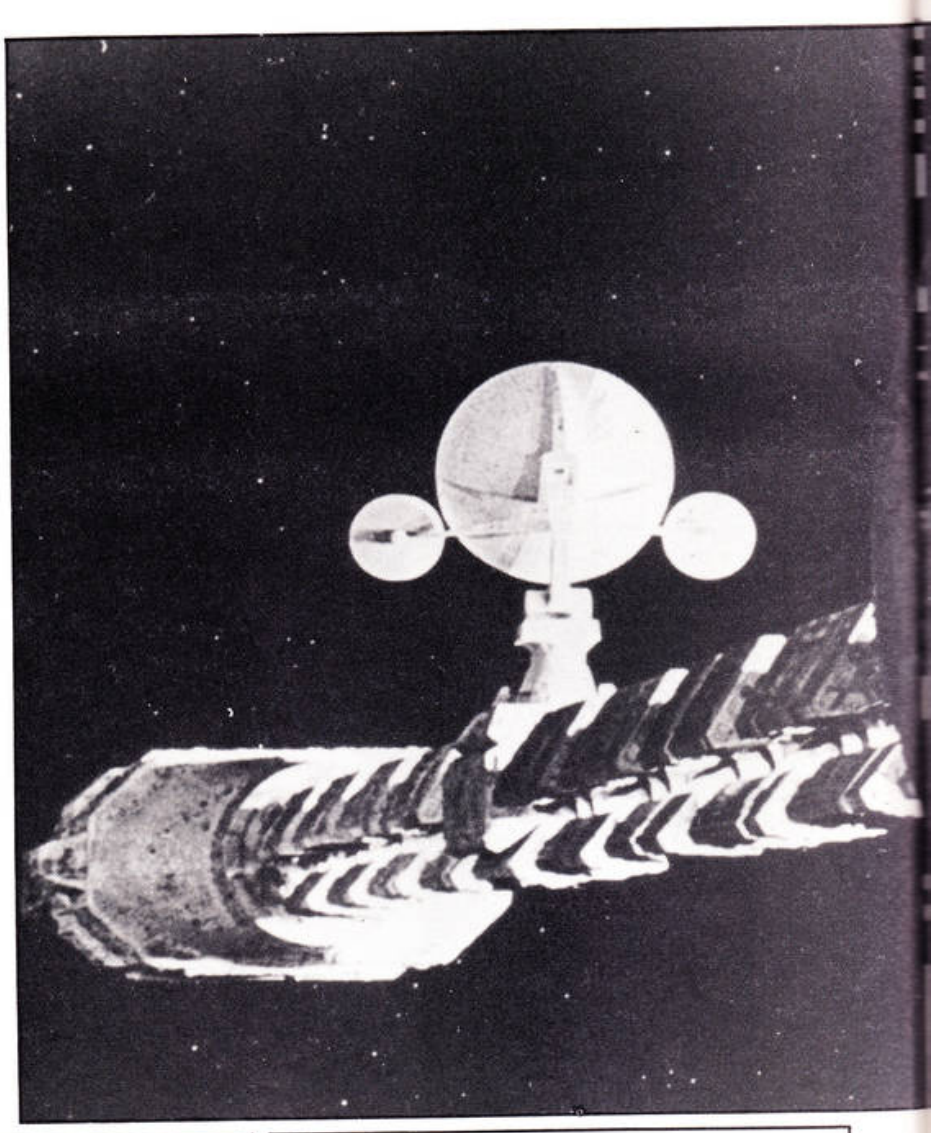
Fig 2b - 48K Machine Code

The object of 'Space Rescue' is to rescue the six astronauts who are stranded on Mars. The player must fly down to the planet's surface, pick up an astronaut and carry him back to the mothership. Between the planet and the Mothership is an asteroid belt. If the ship hits an asteroid it will crash.

The ship is controlled using keys 'Q' and 'W' to move left and right, and key 'P' to apply upward thrust. While thrust is being applied, the ship cannot move horizontally, making ascent more tricky.

After the six men have been picked up, the player must try again, but with more asteroids in the belt.

To fit the program into 16K, I have had to write a separate loader program to enter the data for the machine code and the UDG's.



How to enter

Enter the loader program, RUN it, and enter the two tables of data, first the machine code, then the UDG patterns.

Enter 'NEW' and then type in the actual game listing. Load the code back into memory if it isn't already there and you are ready to play. To SAVE a finished version enter 'GO TO 9100' and start the cassette.

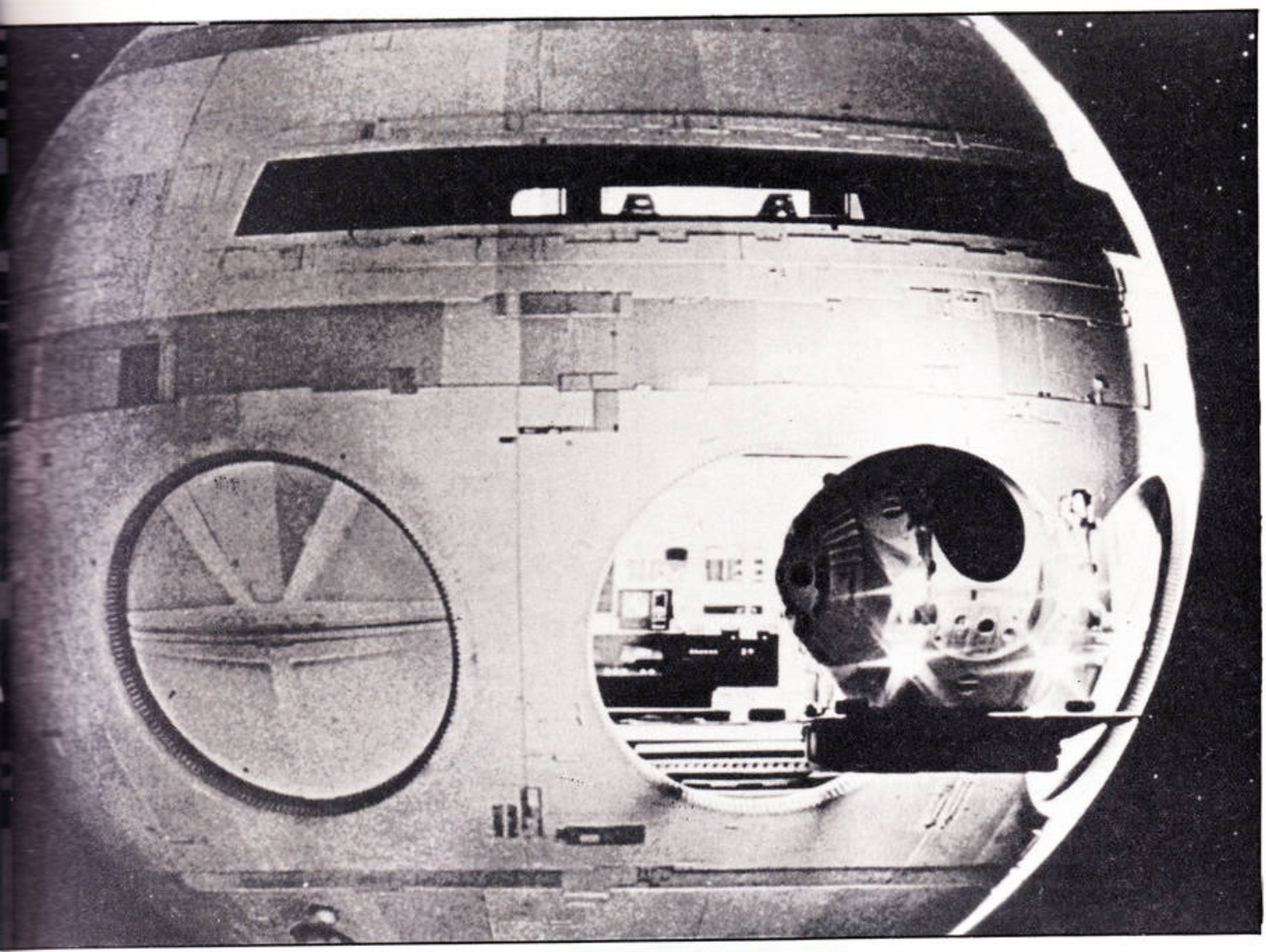
Line-by-line details

20-200	Main game loop
500-530	Collision detect routine
1000-1040	Draw the landscape
1100-1190	Draw the asteroids
1200-1230	Print score, fuel and play level
1300-1390	Initialize variables
1400-1450	End of game routine
1500-1560	Print men at the landing bases
1600-1670	Ship touchdown
1700-1790	Dock with mothership
1800-1880	Level completed
1900-1960	Wait for player to press start
2000-2090	Program control loop
3000-3150	Enter name, then bubblesort hi-scores
4000-4250	Instructions
9000-9040	Load the machine code and graphics data
9100-9130	Save program and code

Variables list

SCORE	Players score
LEVEL	Level of play reached
V,H	Co-ordinates of ship
V1,H1	Previous co-ordinates
CARRY	1 if astronaut carried, 0 if none carried
F	Fuel remaining
TOTAL	Number of men rescued
BASE	1 if lander on left base, 2 for right base
A(2)	Number of men at each base
ATTR	Colour attributes of ship position
H(5)	Array for hi-scores
H$(5,10)	Array for hi-scorer's name
J,K,L	Used in bubblesort for hi-score table

```
 10 REM UDG & mc Input
 20 CLEAR 30999
 30 LET t=0
 35 LET c=0
 40 LET a=31000
 45 PRINT "MACHINE CODE DATA"
 50 INPUT v
 60 POKE a,v
 70 LET a=a+1
 80 LET c=c+v
 85 PRINT TAB t;v;
 90 LET t=t+4
100 IF a<31105 THEN  GO TO 50
110 IF c<>10588 THEN  PRINT '''
FLASH 1;"DATA ERROR !!": BEEP 1
,-10: STOP
120 PRINT '''"DATA ENTERED OK"
130 PRINT ''"USER DEFINED GRAPH
ICS DATA"
140 PRINT
150 LET t=0
160 LET c=0
170 LET a=USR "a"
180 INPUT v
190 POKE a,v
```



SPACE RESCUE

BY GILES TOMAN

See if you can keep a clear head in a crisis and rescue the stranded astronauts.

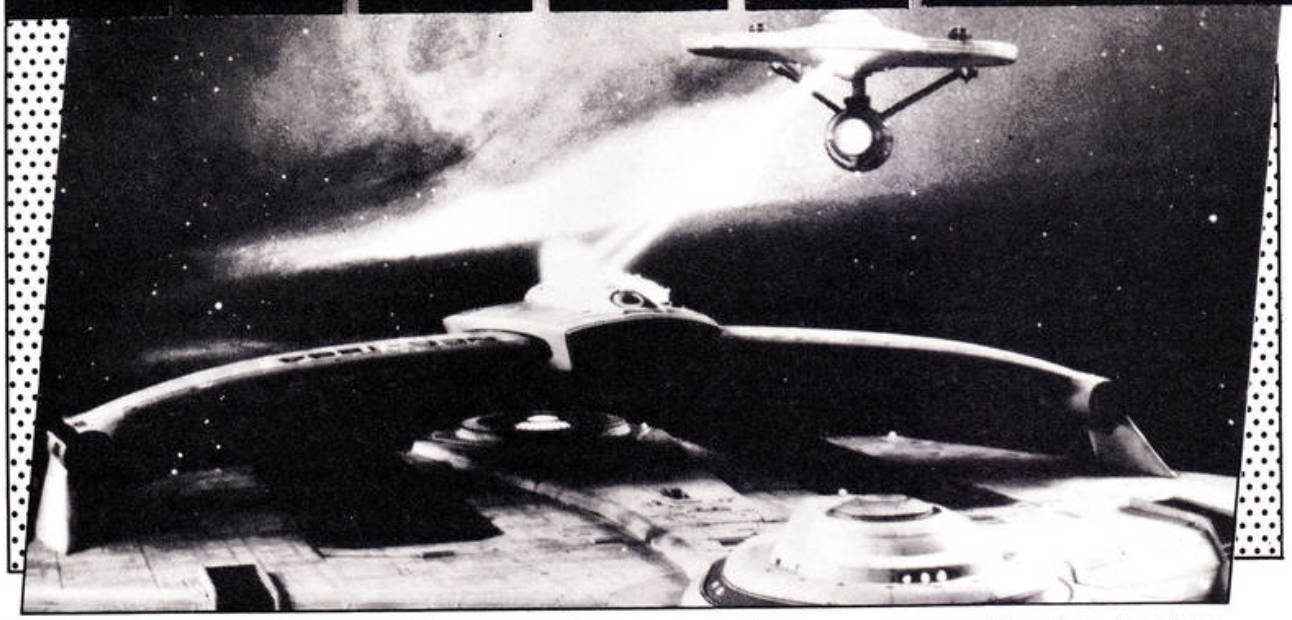
```
 200 LET a=a+1
 210 LET c=c+v
 220 PRINT TAB t;v;
 230 LET t=t+4
 240 IF a<USR "t" THEN  GO TO 18
0
 250 IF c<>21065 THEN  PRINT '''
 FLASH 1;"DATA ERROR !!": BEEP 1
,-10: GO TO 130
 260 PRINT '''"ALL DATA ENTERED O
K"
 270 PRINT ''"PREPARE TO SAVE"
 280 SAVE "SR mc"CODE 31000,105:
 SAVE "SR udg"CODE USR "a",168
```

SPACE RESCUE

```
  10 CLEAR 30999: GO TO 2000
  20 LET a$=INKEY$
  40 LET v1=v: LET h1=h
  50 LET v=v+1-2*(a$="p" AND v>1
 AND f>0): LET h=h+(a$="w" AND h
<31)-(a$="q" AND h>0)
  60 PRINT AT v1,h1;" "
  70 RANDOMIZE USR 31000
  80 IF SCREEN$ (v,h)<>" " OR v>
19 THEN  GO TO 500
  90 PRINT AT v,h; INK 5;"A"
 100 LET f=f-1-1*(a$="p")
 110 IF f<0 THEN  LET f=0
 120 PRINT AT 0,16; INK 3;f;" "
 200 GO TO 20
 500 LET attr=ATTR (v,h)
 510 IF attr=2 OR attr=6 THEN  G
O TO 1400
 520 IF attr=49 THEN  GO TO 1600
 530 IF attr=4 THEN  GO TO 1700
1000 REM -draw landscape-
1010 BORDER 0: PAPER 0: INK 6: C
LS
1020 PRINT AT 21,0; INK 2;"
";AT
20,0;"
";AT 19,0;"
";AT 18,0;"
"
1030 PRINT AT 19,24; INK 1; PAPE
R 6;"BASE";AT 18,6;"BASE"
1035 PRINT AT 2,15; INK 4;"  "
1040 RETURN
1100 REM -print asteroids-
1110 PRINT AT 8,3;"  ";AT 9,3;"
";AT 13,15;"  ";AT 14,15;"  "
1115 IF level=1 THEN  RETURN
1120 PRINT AT 10,20;"  ";AT 11,2
0;"  ";AT 9,10;"  ";AT 12,24;"  "
1130 IF level=2 THEN  RETURN
1140 PRINT AT 11,30;"  ";AT 12,3
0;"  ";AT 13,2;"  "
1150 IF level=3 THEN  RETURN
1160 PRINT AT 14,6;"  ";AT 13,11;
"  ";AT 8,25;"  ";AT 9,25;"  "
1165 IF level=4 THEN  RETURN
1170 PRINT AT 9,14;"  ";AT 15,
21;"  ";AT 14,26;"  "
1190 RETURN
1200 REM -print score etc-
1210 INK 3: PRINT AT 0,0;"SCORE
";score;AT 0,11;"FUEL ";INT f;AT
 0,21;"LEVEL ";level
1220 INK 6
1230 RETURN
1300 REM -initialize vars-
1310 LET score=0
1320 LET level=1
1330 LET v=3: LET h=5
1340 LET carry=0
1350 LET f=1000
1360 DIM a(2)
1370 LET a(1)=3: LET a(2)=3
1380 LET total=0
1390 RETURN
1400 REM -game over-
1410 RANDOMIZE USR 31028
1420 GO SUB 1200
1430 PRINT AT 10,11;"GAME  OVER"
1440 FOR i=1 TO 300: NEXT i
1450 RETURN
1500 REM -print men at bases-
1505 PRINT AT 21,6;"   ";AT 21,2
4;"   "
1510 FOR a=1 TO 2
1520 FOR b=1 TO a(a)
1530 PRINT AT 21,b+5+18*(a=2); I
NK 4;"i"
1540 NEXT b
1550 NEXT a
1560 RETURN
1600 REM -landed-
1610 LET base=1: IF h>20 THEN  L
ET base=2
1620 LET v=v1
1630 IF carry=1 OR a(base)=0 THE
N  GO TO 90
1635 LET carry=1
1640 LET a(base)=a(base)-1
1650 GO SUB 1500
1660 RANDOMIZE USR 31084
1665 LET score=score+5*level
1666 GO SUB 1200
1670 GO TO 90
1700 REM -dock-
1710 LET v=3: LET h=16
1720 IF carry=0 THEN  GO TO 90
1730 LET total=total+1
1740 LET carry=0
1750 PRINT AT 3,16; INK 5;"A"
1760 RANDOMIZE USR 31084: RANDOM
IZE USR 31084
1770 LET score=score+10*level
1780 GO SUB 1200
1785 IF total>=6 THEN  GO SUB 18
00
```

```
1790 GO TO 90
1800 REM -level complete-
1810 GO SUB 1330
1820 CLS
1830 LET level=level+1
1840 GO SUB 1000: GO SUB 1500: G
O SUB 1100
1850 LET score=score+100*level
1860 GO SUB 1200
1870 POKE 31101,35: FOR a=1 TO 8
: RANDOMIZE USR 31084: NEXT a
1875 POKE 31101,43
1880 RETURN
1900 REM -await start key-
1905 LET a=0
1910 INPUT 0
1920 PRINT #1; INVERSE 1;"Press
'S' to start game...      "
1930 FOR i=1 TO 700
1940 IF INKEY$="s" THEN  LET a=1
: RETURN
1950 NEXT i
1960 RETURN
2000 REM -main control loop-
2005 GO SUB 9000
2010 GO SUB 4000
2020 GO SUB 1000
2030 GO SUB 1300
2040 GO SUB 1200
2050 GO SUB 1100
2060 GO SUB 1500
2070 GO SUB 20
2080 GO SUB 3000
2085 IF score>=h(5) THEN  GO SUB
 4190
2090 GO TO 2010
3000 REM -hi score-
3010 IF score<h(5) THEN  RETURN
3020 CLS : PRINT AT 10,11; FLASH
 0;"*WELL DONE*"
3025 BEEP .25,22: BEEP .125,22:
BEEP .125,17: BEEP .25,14: BEEP
.25,10: BEEP .5,22
3030 LET h(5)=score
3035 PRINT AT 21,0;"ENTER YOUR N
AME(10 CHRS MAX)"
3040 INPUT  LINE h$(5)
3050 FOR j=1 TO 4
3060 LET k=j+1
3070 FOR i=k TO 5
3080 LET l=5+k-i
3090 IF h(l)<h(j) THEN  GO TO 31
30
3100 LET a=h(l): LET a$=h$(l)
3110 LET h(l)=h(j): LET h$(l)=h$
(j)
3120 LET h(j)=a: LET h$(j)=a$
3130 NEXT i
3140 NEXT j
3150 RETURN
4000 REM -instructions-
4010 CLS : PRINT TAB 10;"SPACE R
ESCUE"
4020 PRINT '''"INSTRUCTIONS"''"Si
x astronauts are stranded on"''"
Mars.You must pilot the rescue"'
'"ship through the asteroid belt
"''"to the surface and rescue th
em."''"You must then fly them ba
ck to"''"the mothership."''"You
can only carry one astronaut"''"
at a time."
4030 GO SUB 1900
4040 IF a=1 THEN  RETURN
4050 CLS
4060 GO SUB 1000: GO SUB 1300: G
O SUB 1500
4070 PRINT AT 0,10;"SPACE RESCUE
"
4080 PRINT '''''"There are two b
ases,with three"''"astronauts at
 each."
4090 PRINT '"If you try to land
anywhere"''"except the bases,you
 will crash"''"your ship."
4100 GO SUB 1900
4110 IF a=1 THEN  RETURN
4120 CLS
4130 PRINT TAB 10;"SPACE RESCUE"
4140 PRINT '''Keys:"''"LEFT.....
.Q"''"RIGHT.....W"''"THRUST....P
"
4150 PRINT '"Fuel is used contin
uously,but"''"more is used when
'THRUST' is"''"being pressed."
4160 PRINT '"If you run out of f
uel thrust"''"will no longer wor
k,so you"''"cannot continue."
4170 GO SUB 1900
4180 IF a=1 THEN  RETURN
4190 CLS : PRINT TAB 10;"SPACE R
ESCUE"
4200 PRINT '"HI-SCORES:"
4210 PRINT : PRINT
4220 FOR a=1 TO 5: PRINT h$(a);"
 ";h(a): PRINT : NEXT a
4230 GO SUB 1900
4240 IF a=1 THEN  RETURN
4250 GO TO 4000
9000 REM -load code-
9010 DIM h$(5,10): DIM h(5)
9020 FOR a=1 TO 5: LET h$(a)="..
........": NEXT a
9030 LOAD "SR mc"CODE : LOAD "SR
 udg"CODE
9040 RETURN
9100 REM -save program & data-
9110 SAVE "SR" LINE 10
9120 SAVE "SR mc"CODE 31000,105
9130 SAVE "SR udg"CODE USR "a",1
68
```

ASTRO



BY GEORGES BERY

Use your 16K ZX81 to survey the night sky.

What a pleasure, on a clear night, to recognize constellations and stars! This program, written for friends interested in the subject, has been a good conversation piece for other nature lovers. This program is only a threshold . . . once the door is open, beware . . . there is a spell in the night sky.

```
    5 LET Z$="
"
   10 LET A$="
   15 LET F$="
   20 LET B$="
   30 GOSUB 6000
   40 GOSUB 5500
   50 PRINT AT 5,14;F$;AT 10,19;F
$;AT 16,1+;F$
   60 FOR A=0 TO 10
   90 PRINT AT 3,8;B$;AT 3,6;B$;A
T 6,6;B$;AT 7,8;B$;AT 8,9;B$;AT
9,9;B$
  100 PRINT AT 3,8;A$;AT 3,6;A$;A
T 6,6;A$;AT 7,8;A$;AT 8,9;A$;AT
9,9;A$;AT 17,6;A$;AT 2,18;A$
  120 NEXT A
  130 PRINT AT 21,3;"CONSTELLATIO
N?"
  140 LET C$="URSA MAJOR"
  150 INPUT D$
  160 PRINT AT 3,22;D$
  170 IF C$<>D$ THEN PRINT AT 3,2
2;"URSA MAJOR"
  200 LET N=3
  205 LET M=9
  210 GOSUB 5000
  220 LET C$="DHUBE"
  240 INPUT D$
  250 PRINT AT 5,22;C$
  260 IF D$<>C$ THEN PRINT AT 5,2
2;"DHUBE"
  270 PRINT AT N,M;B$
  300 LET N=3
  305 LET M=7
  310 GOSUB 5000
  320 LET C$="MERAK"
  330 INPUT D$
  335 PRINT AT 6,22;C$
  340 IF D$<>C$ THEN PRINT AT 6,2
2;"MERAK"
  350 PRINT AT N,M;B$
  400 LET N=5
  405 LET M=7
  410 GOSUB 5000
  420 LET C$="PHECDA"
  430 INPUT D$
  435 PRINT AT 7,22;C$
  440 IF D$<>C$ THEN PRINT AT 7,2
2;"PHECDA"
  445 PRINT AT N,M;B$
  500 LET N=8
  505 LET M=10
  510 GOSUB 5000
  520 LET C$="ALIOTH"
  530 INPUT D$
  535 PRINT AT 8,22;C$
  540 IF D$<>C$ THEN PRINT AT 8,2
2;"ALIOTH"
  545 PRINT AT N,M;B$
  600 LET N=9
  605 LET M=10
  610 GOSUB 5000
  620 LET C$="ALKAID"
  630 INPUT D$
  635 PRINT AT 9,22;C$
  640 IF D$<>C$ THEN PRINT AT 9,2
2;"ALKAID"
  645 PRINT AT N,M;B$
  700 LET N=17
  705 LET M=7
  710 GOSUB 5000
  720 LET C$="ARCTURUS"
  730 INPUT D$
  735 PRINT AT 12,22;C$
  740 IF D$<>C$ THEN PRINT AT 12,
22;"ARCTURUS"
  745 PRINT AT N,M;B$
```

How To Use It

After a screen of presentation, (don't miss the word "STARS" coming from outside the galaxy and the shooting star) where you can read the simple instructions, you just have to wait to follow the advice appearing on your screen.

When a constellation twinkles or an arrow shines near a star, wait until the word "constellation" or "star" appears at the bottom of the star chart; you have then two possible reactions:

1. You know the answer? You type it followed by N/L. If correct, the answer will be printed at the right side of the star chart in normal video. If NOT correct, the right answer will appear in inverse video. Any easy way to evaluate your knowledge!

or

2. You simply push N/L to see the answer.

It is interesting to note that the names of stars, members of the same constellation, are printed in a block without spacing.

And The Program?

A straight self-explanatory ZX81 BASIC program, which allows easy developments and additions. The addition of an extra star, for instance, only costs 9 short lines (a bargain!).

To "SAVE" the progam, once the typing work is completed, just type "GOTO 7000", start your recorder (on recording position of course) and push N/L. The program is self starting.

References

Various names are given to the same constellation or the same star. For instance, Ursa Major is also called The Dipper, or Alkaid was known as Benetnasch! A choice had to be made. For the presentation of the star charts and the names used in the program, the references are: RUDE STAR FINDER AND IDENTIFIER (HO 2101-C) and AIR ALMANAC (AP 1602).

Last Remarks...

In this program you meet constellations visible only in the Northern hemisphere (around latitude 55). Sorry for the ZX users of the Southern hemisphere!

If you go from your screen to the sky, don't forget that the position of the constellations changes in relation to a ground reference and that certain stars are not always visible.

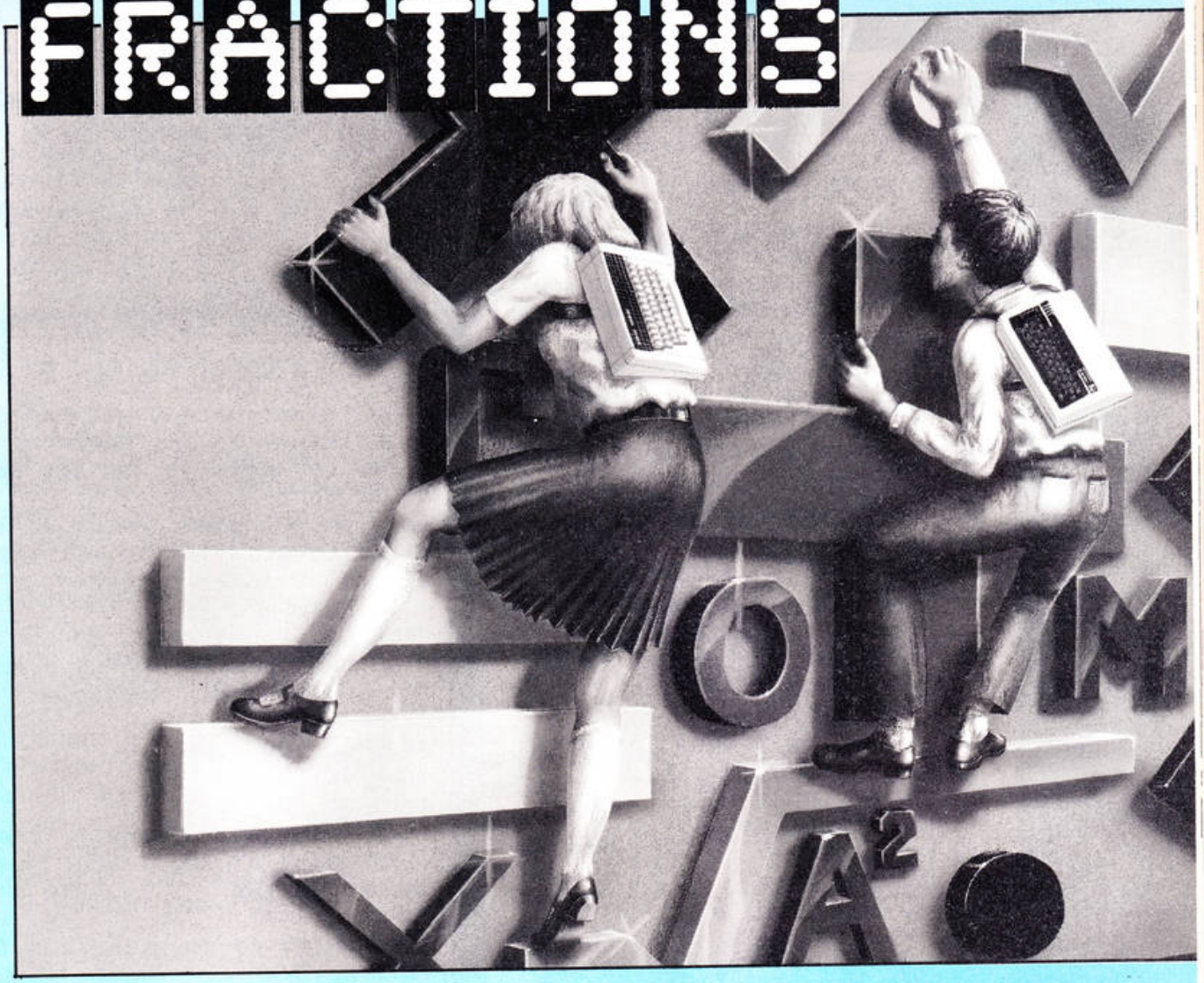
Have a good trip to the stars with your ZX81.

```
 800 LET N=2
 805 LET M=19
 810 GOSUB 5000
 820 LET C$="POLARIS"
 830 INPUT D$
 835 PRINT AT 14,22;C$
 840 IF D$<>C$ THEN PRINT AT 14,
22;"POLARIS"
 845 PRINT AT N,M;B$
 900 PRINT AT 21,0;Z$;AT 21,22;"
 PUSH ""S"""
 910 IF INKEY$<>"S" THEN GOTO 91
0
 920 IF INKEY$="S" THEN GOSUB 55
00
 940 PRINT AT 11,9;A$;AT 12,6;A$
;AT 17,10;A$;AT 17,5;A$;;AT 14,7
;A$;AT 14,8;A$;AT 14,9;A$;AT 11,
1;A$;AT 2,3;A$;AT 2,14;A$;AT 3,1
5;A$;AT 7,19;A$;AT 16,17;A$
 945 FOR A=0 TO 10
 950 PRINT AT 11,9;B$;AT 12,6;B$
;AT 17,5;B$;AT 17,10;B$;AT 14,7;
B$;AT 14,8;B$;AT 14,9;B$
 960 PRINT AT 11,9;A$;AT 12,6;A$
;AT 17,5;A$;AT 17,10;A$;AT 14,7;
A$;AT 14,8;A$;AT 14,9;A$
 970 NEXT A
 980 PRINT AT 7,5;F$;AT 8,11;F$;
AT 18,19;F$
 990 PRINT AT 21,3;"CONSTELLATIO
N?"
1000 LET C$="ORION"
1010 INPUT D$
1015 PRINT AT 1,22;C$
1020 IF D$<>C$ THEN PRINT AT 1,2
2;"ORION"
1100 LET N=11
1105 LET M=10
1110 GOSUB 5000
```

```
1115 LET C$="BETELGEUSE"
1125 INPUT D$
1130 PRINT AT 4,22;C$
1135 IF D$<>C$ THEN PRINT AT 4,2
2;"BETELGEUSE"
1140 PRINT AT N,M;B$
1200 LET N=12
1205 LET M=7
1210 GOSUB 5000
1215 LET C$="BELLATRIX"
1225 INPUT D$
1230 PRINT AT 5,22;D$
1235 IF D$<>C$ THEN PRINT AT 5,2
2;"BELLATRIX"
1240 PRINT AT N,M;B$
1300 LET N=17
1305 LET M=6
1310 GOSUB 5000
1315 LET C$="RIGEL"
1325 INPUT D$
1330 PRINT AT 6,22;C$
1335 IF D$<>C$ THEN PRINT AT 6,2
2;"RIGEL"
1340 PRINT AT N,M;B$
1400 LET N=11
1405 LET M=2
1410 GOSUB 5000
1415 LET C$="ALDEBARAN"
1425 INPUT D$
1430 PRINT AT 8,22;C$
1435 IF D$<>C$ THEN PRINT AT 8,2
2;"ALDEBARAN"
1440 PRINT AT N,M;B$
1500 LET N=2
1505 LET M=4
1510 GOSUB 5000
1515 LET C$="CAPELLA"
1520 INPUT D$
1525 PRINT AT 10,22;C$
1530 IF D$<>C$ THEN PRINT AT 10,
```

```
22;"CASTOR"
1540 PRINT AT N,M;B$
1600 LET N=2
1605 LET M=15
1610 GOSUB 5000
1615 LET C$="CASTOR"
1625 INPUT D$
1630 PRINT AT 12,22;C$
1635 IF C$<>D$ THEN PRINT AT 12,
22;"CASTOR"
1640 PRINT AT N,M;B$
1700 LET N=3
1705 LET M=16
1710 GOSUB 5000
1715 LET C$="POLLUX"
1725 INPUT D$
1730 PRINT AT 13,22;C$
1735 IF C$<>D$ THEN PRINT AT 13,
22;"POLLUX"
1740 PRINT AT N,M;B$
1800 LET N=7
1805 LET M=20
1810 GOSUB 5000
1815 LET C$="PROCYON"
1825 INPUT D$
1830 PRINT AT 15,22;C$
1835 IF D$<>C$ THEN PRINT AT 15,
22;"PROCYON"
1840 PRINT AT N,M;B$
1900 LET N=16
1905 LET M=18
1910 GOSUB 5000
1915 LET C$="SIRIUS"
1925 INPUT D$
1930 PRINT AT 17,22;C$
1935 IF C$<>D$ THEN PRINT AT 17,
22;"SIRIUS"
1940 PRINT AT N,M;B$
1950 PRINT AT 21,0;Z$;AT 21,22;"
PUSH ""S"""
1955 IF INKEY$<>"S" THEN GOTO 19
55
1960 GOSUB 5500
2000 PRINT AT 3,4;F$;AT 7,7;F$;A
T 9,8;F$;AT 11,9;F$;AT 12,3;F$;A
T 17,1;F$;AT 19,1;F$;AT 18,20;F$
;AT 6,2;A$;AT 8,10;A$;AT 18,1;A$
2005 FOR A=0 TO 10
2010 PRINT AT 0,20;B$;AT 1,19;B$
;AT 2,20;B$;AT 3,19;B$;AT 4,21;B
$
2015 PRINT AT 0,20;F$;AT 1,19;F$
;AT 2,20;A$;AT 3,19;F$;AT 4,21;F
$
2020 NEXT A
2030 PRINT AT 21,3;"CONSTELLATIO
N?"
2035 LET C$="CASSIOPIA"
2040 INPUT D$
2045 PRINT AT 1,22;"CASSIOPEIA"
2050 IF C$<>D$ THEN PRINT AT 1,2
2;"CASSIOPEIA"
2100 LET N=0
2105 LET M=21
2110 GOSUB 5000
2115 LET C$="CAPH"
2120 INPUT D$
2125 PRINT AT 3,22;C$
2130 IF C$<>D$ THEN PRINT AT 3,2
2;"CAPH"
2135 PRINT AT N,M;B$
2200 LET N=1
2210 LET M=20
2215 GOSUB 5000
2220 LET C$="SHEDIR"
2225 INPUT D$
2230 PRINT AT 4,22;C$
2235 IF C$<>D$ THEN PRINT AT 4,2
     2;"SHEDIR"
2240 PRINT AT N,M;B$
2300 LET N=8
2305 LET M=11
2310 GOSUB 5000
2315 LET C$="DENEB"
2320 INPUT D$
2325 PRINT AT 6,22;C$
2330 IF C$<>D$ THEN PRINT AT 6,2
2;"DENEB"
2335 PRINT AT N,M;B$
2400 LET N=18
2405 LET M=2
2410 GOSUB 5000
2415 LET C$="ALTAIR"
2420 INPUT D$
2425 PRINT AT 8,22;C$
2430 IF C$<>D$ THEN PRINT AT 8,2
2;"ALTAIR"
2435 PRINT AT N,M;B$
2500 LET N=6
2505 LET M=3
2510 GOSUB 5000
2515 LET C$="VEGA"
2520 INPUT D$
2525 PRINT AT 10,22;C$
2530 IF C$<>D$ THEN PRINT AT 10,
22;"VEGA"
2535 PRINT AT N,M;B$
2545 PRINT AT 21,1;"ANOTHER GO?
Y/N "
2550 IF INKEY$="Y" THEN GOTO 40
2555 IF INKEY$="" THEN GOTO 2550
2560 GOSUB 5500
2570 PRINT AT 10,1;"NOW THE SK
ES YOURS"
2575 STOP
5000 PRINT AT 21,0;Z$
5010 FOR A=0 TO 20
5020 PRINT AT N,M;B$;AT N,M;"*"
5030 NEXT A
5040 PRINT AT 21,2;"STAR ?"
5050 RETURN
5500 FOR A=21 TO 0 STEP -1
5510 PRINT AT A,0;"
                    "
5520 NEXT A
5530 RETURN
6000 FOR A=15 TO 0 STEP -1
6010 FOR B=15 TO 21
6020 PRINT AT A,0;"
                              "
6030 NEXT A
6040 PRINT AT B,0;"
                              "
6050 NEXT B
6060 FOR A=0 TO 10
6070 PRINT AT 7,4;"*";AT 7,4;"*"
6080 NEXT A
6090 PRINT AT 6,5;"**"
6100 PRINT AT 6,5;"*"
6110 PRINT AT 5,6;"*"
6120 PRINT AT 5,6;"****"
6130 FOR A=0 TO 5
6140 PRINT AT 4,7;"*"
6150 PRINT AT 4,7;"STARS"
6160 NEXT A
6180 FOR A=0 TO 30 STEP 4
6190 PRINT AT 10,A;"*";AT 10,A;"
*"
6200 NEXT A
6220 PRINT AT 11,6;F$;AT 7,12;F$
;AT 3,19;F$;AT 5,20;F$;AT 5,23;F
$;AT 3,24;F$
6230 PRINT AT 17,1;"WHEN ASKED,
ENTER YOUR ANSWER."
6240 PRINT AT 18,1;"A WRONG ANSW
ER IS REPRINTED IN INVERSE VIDE
O."
6250 PRINT AT 20,1;"PUSH N/L FOR
 UNKNOWN ANSWER."
6255 FOR A=0 TO 80
6256 NEXT A
6260 PRINT AT 14,14;"PUSH "S" TO
 START"
6270 IF INKEY$<>"S" THEN GOTO 62
70
6280 RETURN
7000 SAVE "STARS"
7005 RUN
```

FRACTIONS



Teach fractions to your children with the help of this program.

BY PAUL ESTCOURT

The program listed is an introduction to the subject and is one of 4 programs I have written. The other 3 become progressively more complicated.

I bought a tape on this topic to help my daughter who was then 9 years old. Unfortunately, we were very disappointed, so I set about to write my own. I hope readers find my introductory program helpful.

The features are:

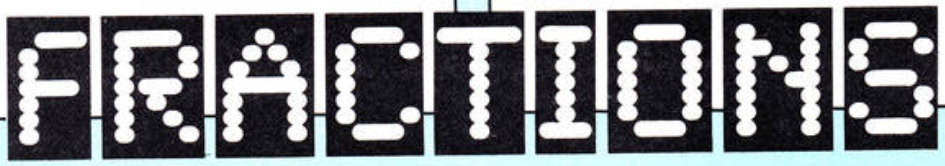
(a) Diagrams
(b) Simple addition and subtraction
(c) Examples and Questions
(d) Sound to reflect whether the answer is right or wrong
(e) Score at end of program and each section
(f) Runs on 16K or 48K.

A breakdown of the program is as follows:

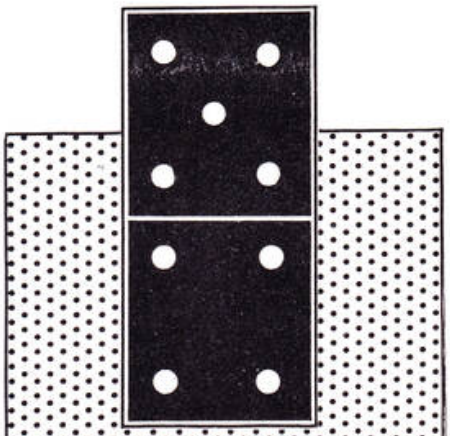
10	Set counters for number of questions and those answered correctly.
20	Set Capitals Lock to ease testing input.
160	Draw Circle and Square.
400	You are asked how many parts the shapes are to be divided up into. Enter the number between 2 and 360. This is the only part of the program where the number may be greater than 9.
401	Test if entry valid.
412	You are shown how to write the fractions.
415	The shapes are divided up and shaded. Note: Delete line 6130 if you want to see the circle shaded. However, if you have chosen to divide the shape into a large number of parts the program may take a long time to shade the circle.
1050	Chance to see another example. If so, program goes to line having the value given to "label".
2000-2040	You are set questions on simple fractions.
2330-4024	Examples/Questions on adding fractions with the same denominator. Note: No attempt is made dealing with cancelling. This is left to the later programs.
4080-4170	Examples/Questions on subtracting fractions.
4175-4180	End of program. Score shown.

```
  10 LET sc=0: LET tot=0: LET go
es=1: LET goes1=0
  20 POKE 23658,0
  30 LET ink=9
  50 BORDER 6: PAPER 6: INK 9: C
LS
 100 PRINT AT 0,16; PAPER 4;"FRA
CTIONS"
 150 PRINT AT 1,16;"
   "
 160 GO SUB 5999
 400 INPUT  PAPER 4; INK 9;"HOW
MANY PARTS SHALL WE MAKE?"; LINE
 p$
 401 GO SUB 6999
 402 IF legal=1 THEN  GO TO 400
 403 LET PARTS=VAL P$
 405 LET no=1
 412 PRINT AT 2,16;"THERE ARE ";
p$;AT 4,16;" PARTS AND EACH ";AT
 6,16;"MAY BE WRITTEN ";AT 8,18;
"AS "; BRIGHT 1;"1/";p$; BRIGHT
0;" OR "; BRIGHT 1;"1";AT 9,28;"
_";AT 11,28;p$
 413 IF VAL p$>9 THEN  PRINT AT
9,29;"_"; BRIGHT 0
 415 GO SUB 6049
1040 LET label=30
1050 GO SUB 6199
2000 GO SUB 6549
2010 GO SUB 5999
2015 PRINT AT 1,15;"WHAT FRACTIO
N IS";AT 2,15;" THE SHADED AREA
";AT 3,15;"OF THE WHOLE?"
2020 LET PARTS=INT (RND*5)+2
2030 GO SUB 6049
2040 GO SUB 6249
2200 LET label=2010: GO SUB 6349
2330 CLS : PRINT  PAPER 2;"LETS
TRY ADDING FRACTIONS": PAUSE 100
2332 GO SUB 6399
3420 PRINT AT 12,5;no;"/";PARTS
3430 LET label=2332: GO SUB 6199
3432 GO SUB 6549
3440 GO TO 4020
4020 GO SUB 6399
4022 LET sub=0: GO SUB 4029
4024 GO TO 4079
4030 PRINT AT 15,13;"/?";AT 15,1
2; BRIGHT 1; FLASH 1;"?"
4040 PRINT #1;AT 0,0; PAPER 4; F
LASH 1;"ENTER NUMBER ON "; PAPER
 3; INK 9;"TOP"; PAPER 4;" OF FR
ACTION"
4041 PAUSE 0: LET p$=INKEY$: GO
SUB 7000
4042 IF legal=1 THEN  GO TO 4040
4043 LET no1=VAL p$: LET goes=go
es+1
4044 PRINT AT 15,12;no1;"/"; BRI
GHT 1; FLASH 1;"?"
4050 PRINT #1;AT 0,0; PAPER 4; F
LASH 1;"ENTER NUMBER ON "; PAPER
 1; INK 9;"BOTTOM"; PAPER 4;" OF
        FRACTION"
4051 PAUSE 0: LET P$=INKEY$: GO
SUB 7000
4052 IF legal=1 THEN  GO TO 4050
4053 LET no2=VAL p$
4055 PRINT AT 15,14;no2
4056 IF sub=1 THEN  LET no=PARTS
-no
4060 IF no1=no AND no2=PARTS THE
N  PRINT AT 16,16; PAPER 6;"CORR
ECT": LET sc=sc+1: GO SUB 5000:
GO TO 4075
4070 GO SUB 5500: PRINT AT 16,16
;"THE CORRECT";AT 17,16;"ANSWER
IS";AT 18,16;no;"/";PARTS
4075 RETURN
4080 LET label=4020: GO SUB 6349
4100 CLS : PRINT  PAPER 2;"LETS
TRY SUBTRACTING FRACTIONS": PAUS
E 100
4105 GO SUB 4109
4107 GO TO 4142
4110 GO SUB 6399
4120 PRINT AT 10,5;no;"/";PARTS
4130 PRINT AT 12,1;"WHAT MUST BE
 ADDED TO THE       SHADED AREA T
O MAKE '1'?"
4140 PRINT AT 15,2;"ANSWER IS ";
4141 RETURN
4142 LET no=PARTS-no
4143 PRINT AT 15,12;no;"/";PARTS
4145 LET label=4105: GO SUB 6199
4150 GO SUB 6549
4155 GO SUB 4109
4165 LET sub=1: GO SUB 4029
4170 LET label=4150: GO SUB 6349
4175 REM END OF LESSON****
4177 CLS : PRINT AT 15,0; PAPER
1; INK 7;"YOUR FINAL SCORE IS ";
tot;" QUESTIONS RIGHT OUT OF ";g
oes1;" THAT IS ";INT (tot*100/go
es1);"%."
4180 PRINT  PAPER 4; INK 9;AT 7,
11; FLASH 1;"WELL DONE!!!": GO S
UB 5000: GO SUB 5510: GO SUB 552
0: PAUSE 100: CLS : PRINT AT 7,1
1; FLASH 1;"BYE FOR NOW": STOP
4999 REM BEEP****
5000 FOR n=30 TO 50: BEEP .1,n:
NEXT n: RETURN
5500 FOR n=50 TO 30 STEP -1: BEE
P .1,n: NEXT n: RETURN
5510 FOR n=30 TO 50 STEP 2: BEEP
 .1,n: NEXT n: RETURN
5520 FOR n=30 TO 50 STEP 4: BEEP
```

```
.1,n: NEXT n: RETURN
5999 REM DRAW SHAPES****
6000 INK ink: CIRCLE 30,150,25
6010 PLOT 64,125: DRAW 50,0: DRA
W 0,50: DRAW -50,0: DRAW 0,-50
6020 INK 9: RETURN
6049 REM DIVIDE INTO N PARTS****
6050 PLOT 30,150: DRAW 25,0
6060 FOR N=1 TO PARTS
6070 PLOT 30,150: DRAW 25*COS (2
*N*PI/PARTS),25*SIN (2*N*PI/PART
S)
6080 PLOT (64+50/PARTS*N),125: D
RAW 0,50
6090 NEXT N
6100 FOR n=1 TO 50/PARTS*no
6110 PLOT 64+n,125: DRAW 0,50
6120 NEXT n
6130 RETURN : REM THIS STATEMENT
 MAY BE DELETED TO SHADE THE CIR
CLE**********
6140 FOR n=1 TO (360/PARTS*no)
6150 PLOT 30,150: DRAW 25*COS (n
*PI/180),25*SIN (n*PI/180)
6160 NEXT n
6180 RETURN
6199 REM ANOTHER EXAMPLE?****
6200 PRINT #1;AT 0,0; PAPER 4; I
NK 9; FLASH 1;"ANOTHER EXAMPLE?-
ENTER 'Y'OR 'N'"
6205 PAUSE 0: LET A$=INKEY$
6210 IF A$="Y" THEN  CLS : GO TO
 label
6220 CLS
6230 RETURN
6249 REM ENTER NUMBERS INTO FRAC
TIONS****
6250 PRINT #1;AT 0,0; PAPER 4; F
LASH 1;"ENTER NUMBER ON "; PAPER
 3; INK 9;"TOP"; PAPER 4;" OF FR
ACTION"
6255 PAUSE 0: LET P$=INKEY$
6256 GO SUB 7000: IF legal=1 THE
N  GO TO 6250
6257 LET A=VAL P$
6260 PRINT #1;AT 0,0; PAPER 4; F
LASH 1;"ENTER NUMBER ON "; PAPER
 1; INK 9;"BOTTOM"; PAPER 4;" OF
          FRACTION"
6262 PAUSE 0: LET P$=INKEY$
6263 GO SUB 7000: IF legal=1 THE
N  GO TO 6260
6264 LET B=VAL P$
6270 PRINT AT 6,16;"YOU ENTERED
";A;"/";B
6280 IF A=1 AND B=PARTS THEN  PR
INT AT 8,16; PAPER 6;"CORRECT":
LET sc=sc+1: GO SUB 5000: GO TO
2200
6290 GO SUB 5500: PRINT AT 8,16;
"THE CORRECT";AT 9,16;"ANSWER IS
 ";AT 10,20;"1/";PARTS: PAUSE 30
0
6300 RETURN
6349 REM TRY AGAIN?****
6350 PRINT #1;AT 0,0; PAPER 4; I
NK 9; FLASH 1;"DO YOU WISH TO TR
Y AGAIN -ENTER 'Y' OR 'N'"
6355 PAUSE 0: LET B$=INKEY$
6360 IF b$="Y" THEN  CLS : LET g
oes=goes+1: GO TO label
6365 PRINT AT 20,0; PAPER 1; INK
 7;"YOU HAVE GOT ";sc;" QUESTION
S RIGHT"''"OUT OF ";goes;". THAT
IS ";INT (sc*100/goes);"%."
6366 LET tot=tot+sc: LET sc=0: L
IS ";INT (sc*100/goes);"%."
6366 LET tot=tot+sc: LET sc=0: L
ET goes1=goes1+goes: LET goes=0
6367 PRINT #1; FLASH 1; PAPER 1;
 INK 7;AT 0,0;"PRESS ANY KEY TO
CONTINUE
                ": PAUSE 0
6370 RETURN
6399 REM DIVIDE INTO N RANDOM PA
RTS****
6400 LET label=6400: CLS : GO SU
B 6000
6410 PRINT AT 1,16;"THE SHADED A
REA ";AT 2,25;"IS"
6420 LET PARTS=INT (RND*7)
6430 IF PARTS<3 THEN  GO TO 6420
6440 LET no=INT (RND*PARTS)
6450 IF no>=PARTS OR no<2 THEN
GO TO 6440
6460 FOR n=1 TO no
6470 PRINT AT 9,(5*n);"1/";PARTS
;"+";
6480 NEXT n
6490 PRINT CHR$ 8;" ="
6500 GO SUB 6050
6510 RETURN
6549 REM EXERCISES****
6550 CLS : PRINT  PAPER 2;"NOW T
RY THE FOLLOWING QUESTIONS": PAU
SE 100: CLS
6560 RETURN
6999 REM LEGAL INPUT
7010 LET legal=0
7020 IF p$<"0" OR p$>"9" THEN  L
ET legal=1: RETURN
7030 IF VAL p$<>INT VAL p$ THEN
 LET legal=1: RETURN
7040 RETURN
```

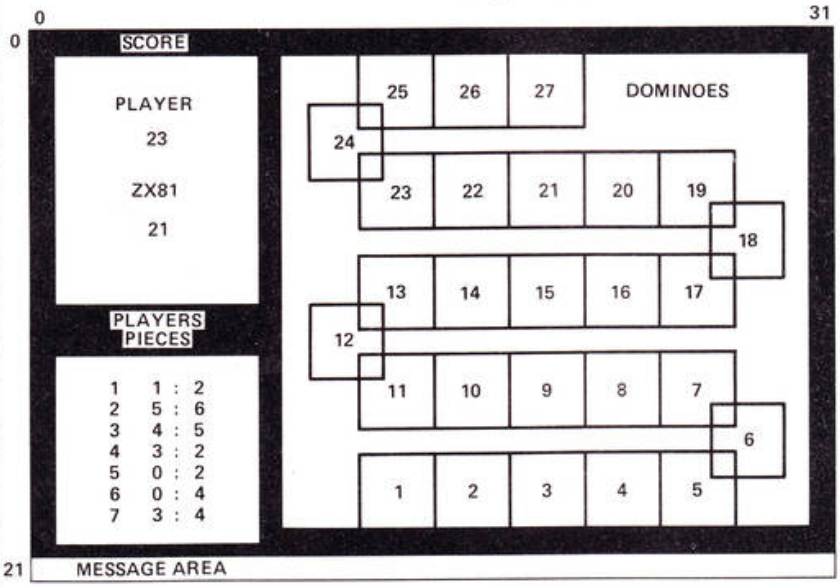


DOMINO



Before you think that you're seeing spots before your eyes, never fear — just enjoy this game of dominoes, which is for the ZX81 plus 16K RAM fitted.

Fig.3. Screen Map showing display of up to 27 dominoes



How about a game of Dominoes..... you against the ZX81?

The computer has an impeccable ability to choose it's highest scoring domino each time. You, on the other hand, must rely on your own mathematical ability.... but you do have the advantage of a human's tactical guile.

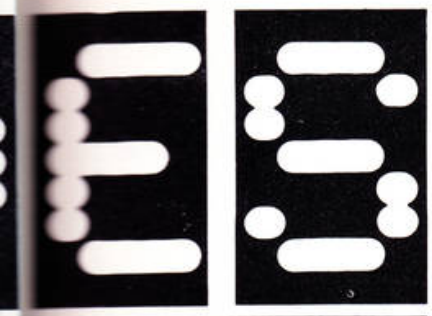
You have the option to play 5's & 3's or 6's 8 4's and you can elect to leave two random dominoes out of the game if you wish to inhibit your powers of deduction and increase the challenge.

Good luck!

Program Notes

LOAD "DOM" and then RUN. You are first greeted with a set of instructions (from line 9000) which give the ground rules for playing this version of Dominoes. During the sequence you are asked if you want to play 5's & 3's or 6's & 4's and the method of scoring is explained. (Deciding which domino will give the best score provides useful long division practice as an educational bonus). You are finally asked if you wish to leave 2 random dominoes unused. This will impede logical deduction at the end of the game.

The dominoes are loaded into memory in increasing order as D$ (line 9530) and each



BY NORMAN BROOKS

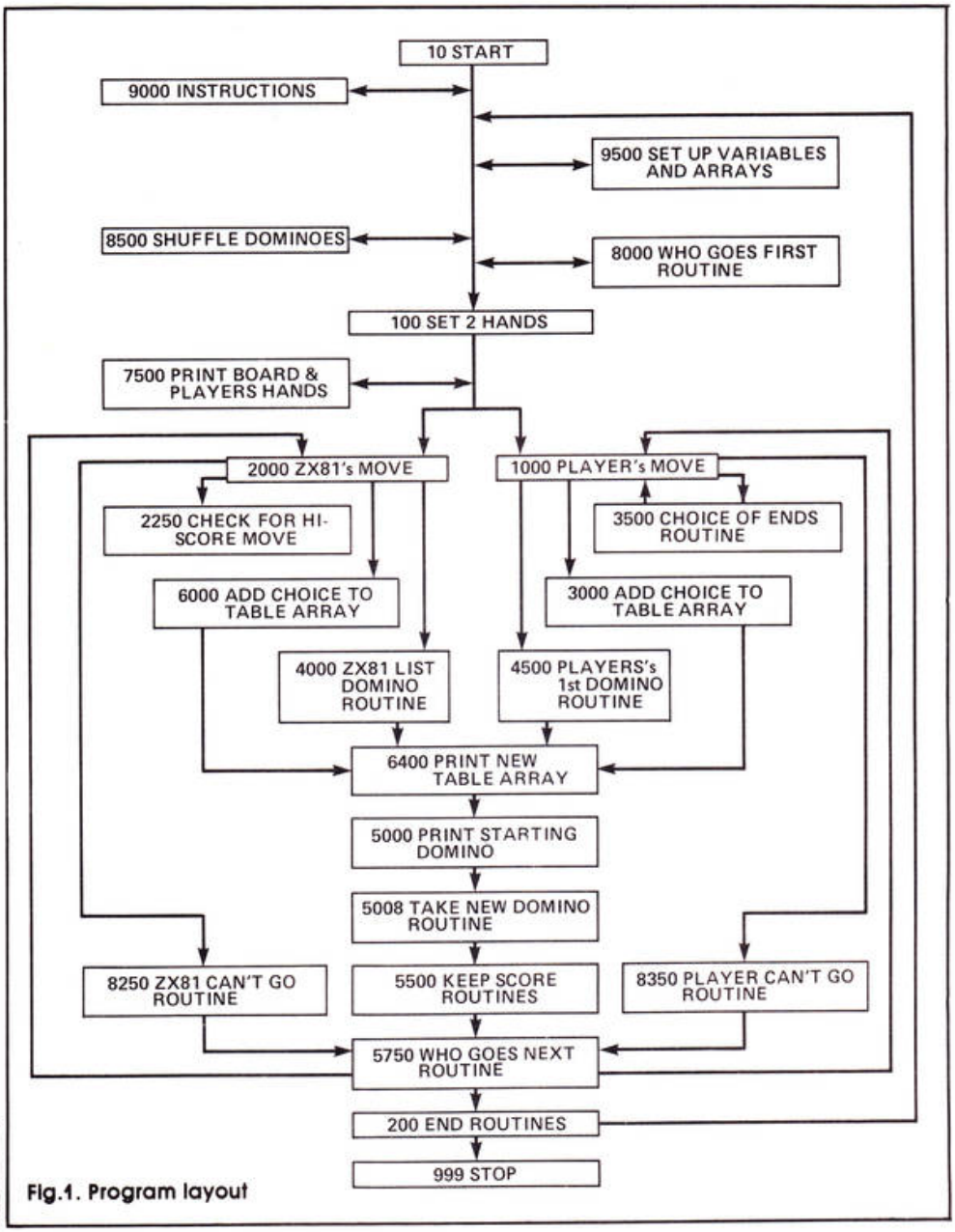
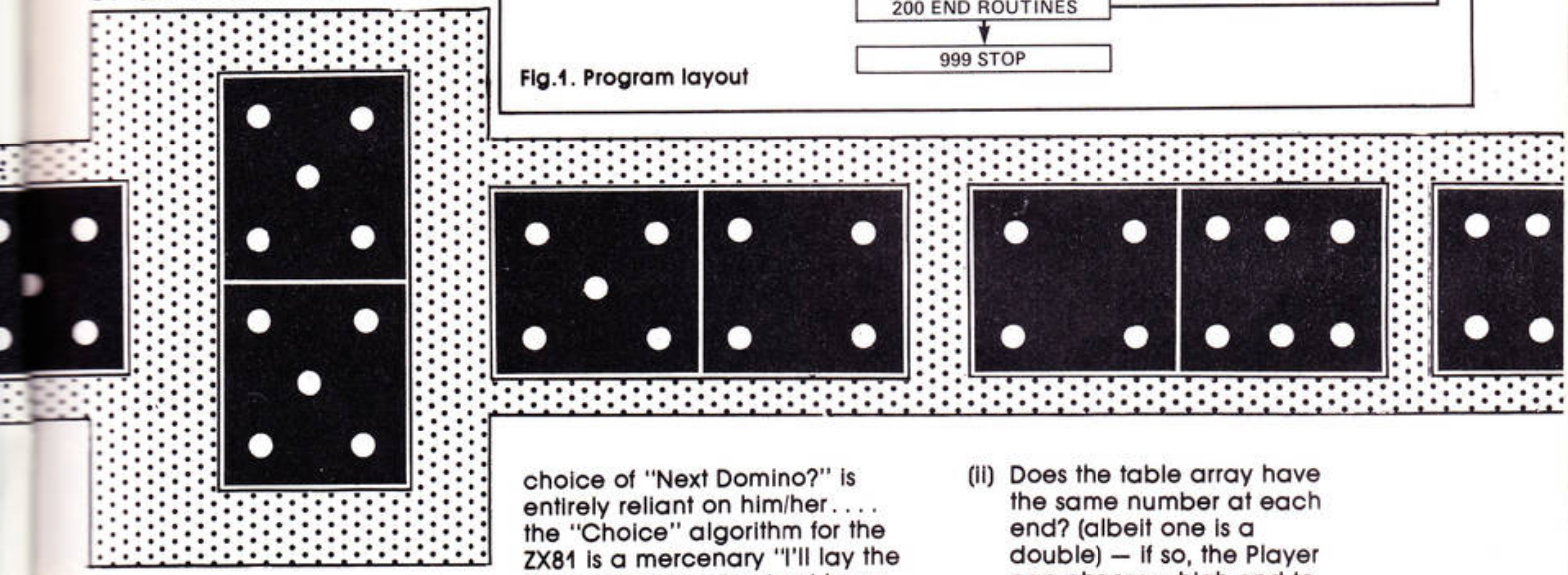


Fig.1. Program layout

domino is 3 characters long, the middle one always being a colon. The shuffle Routine (from line 8500) reads D$ in random 3 character blocks to set up A$(28,3) until all dominoes have been read once. This then is the shuffled pack. The player is given the first 7 as P$(14,3), the ZX81 the next 7 as Z$(14,3) and this leaves 14 dominoes (or 12 if 2 are to remain unused) for each player to draw on as they lay. Note that the first dimension of P$ and Z$ is 14 and not 7. The extra 7 locations are used to store the mirror image of each domino held. This helps in checking procedures later. I have chosen to store the array of laid dominoes (the table array) as B$(3,81). This allows "doubles" to be laid crosswise and a maximum of 27 dominoes to be accommodated on the screen. (see fig. 3)



Whose choice?

"Who Goes First" is decided by a simple dice throwing routine (from line 8000). The Player's choice of "Next Domino?" is entirely reliant on him/her.... the "Choice" algorithm for the ZX81 is a mercenary "I'll lay the highest scoring domino I have and never mind the consequences." The next problem is one of IF/THEN logic. My solution is —

(i) Is it the first domino of the game? — if so forget any checking routines.

(ii) Does the table array have the same number at each end? (albeit one is a double) — if so, the Player can choose which end to lay.

(iii) Do the numbers either end of the table array match the two numbers on the chosen domino? — if so, again the Player can choose which end to lay. (Line 1060 performs this

function and is the longest I have ever written so take care when typing!)

(iv) Does the 1st no. on the domino match the last no. on the table?
— if so, add the chosen domino to the table array with the correct orientation. (allowing for doubles of course).

(v) Finally, if none of the above hold true, it is an illegal move and the Player is penalised 1 point. (The ZX81 cannot make an illegal move!)

Printing the new table array presents the next problem. Due to screen size I can't place the first domino in the centre and work outwards from it. Hence I start the chain at the bottom left and snake it upto the screen on subsequent moves. (from line 6400). Dominoes are either added to the end of the chain or the whole array is moved round 3 characters and then the new domino is inserted at the front end. This latter routine is time consuming after the 5th domino has been laid as I need to use nested loops for the "right to left" printing routines from then on. I have managed to speed things up by creating S$(3,15) as the reverse of B$(1 to 3,19 to 33) and B$(1 to 3, 55 to 69) during the routine and then printing S$ rather than printing B$ backwards directly. (i.e. LET statements are executed faster than PRINT statements). This and the "I'm just moving the pieces round" message have made a marked improvement to the running of the program.

End of move

Each move is rounded off with (a) the taking of a replacement domino from the unused pack (if it still exists), (b) calculation and display of the new score, (c) a check to see if the game is ended (ie someone has used all their dominoes or neither player can go) and (d) go to the next player's turn. Line 5980 LET G=(G=0) provides an efficient way in Basic to alternate the "Who Goes Next" flag between 0 and 1.

At the end of the game, the Player can carry the scores into another round or terminate play with the ZX81 declaring who has won. (from line 200).

Major Variables

Z	Computer's score
P	Player's score
G	Flag for "Who Goes Next" (0=ZX81 1=Player)
D	Number of dominoes taken from unused pack
D1	Player's dice throw
D2	ZX81 dice throw
T	Number of dominoes on the table
N	ZX81/Player's choice of "Next Domino"
HS	ZX81 Hi-Score on each move
N1	Number of ZX81 domino to be laid if no score possible
S	Used to computer ZX81
S1 to S5	Hi-Score
A	Flag sensing computer can't go (ie A=1)
B	Flag sensing Player can't go (ie B=1)
V	Sets orientation of Hi-Score domino
V1	Used to set up V
C	Flag sensing which end of chain takes new move (Top=1)
K & L	Scoring method (5 & 3 or 6 & 4)
E	Choice to leave 2 dominoes unused (27=yes, 29=no)
D$	String of all dominoes in increasing order
A$(28,3)	Array of shuffled dominoes
Z$(14,3)	Computer's hand of 7 dominoes and their reverse
P$(14,3)	Player's hand of 7 dominoes and their reverse
B$(3,81)	Array of dominoes on table which is filled with inverse spaces at the start of a game
S$(3,15)	Used to speed printing B$

(C$, R$, R and I aid the running of the program.)

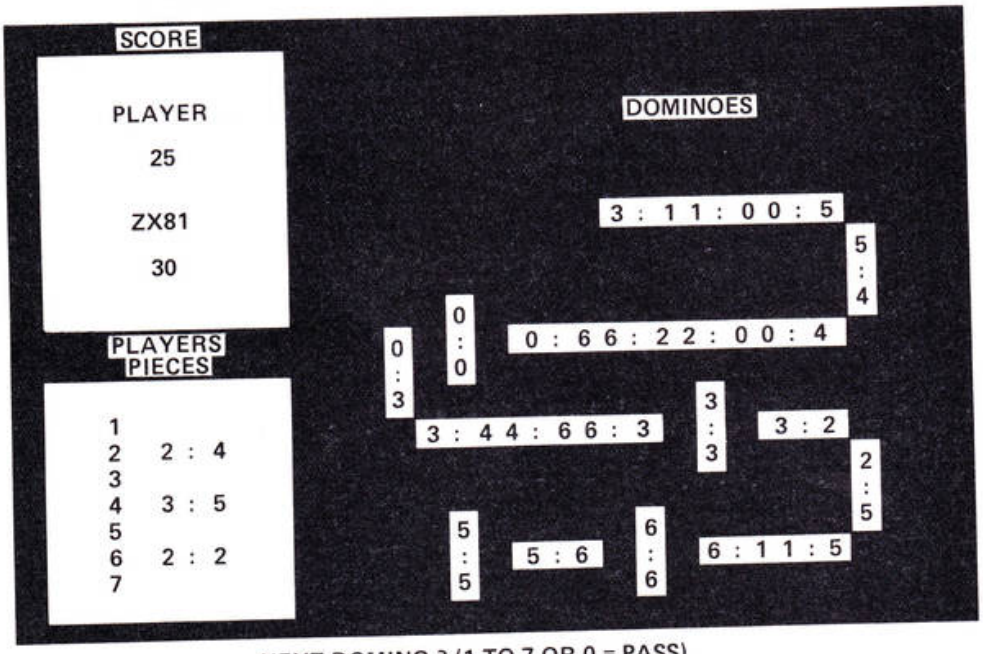
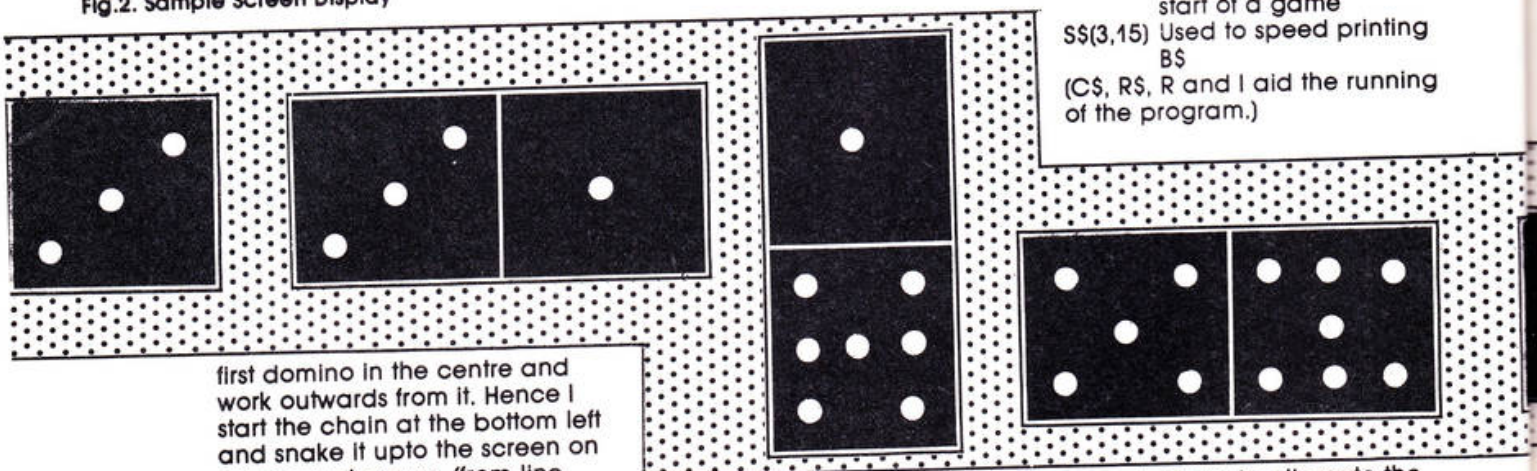
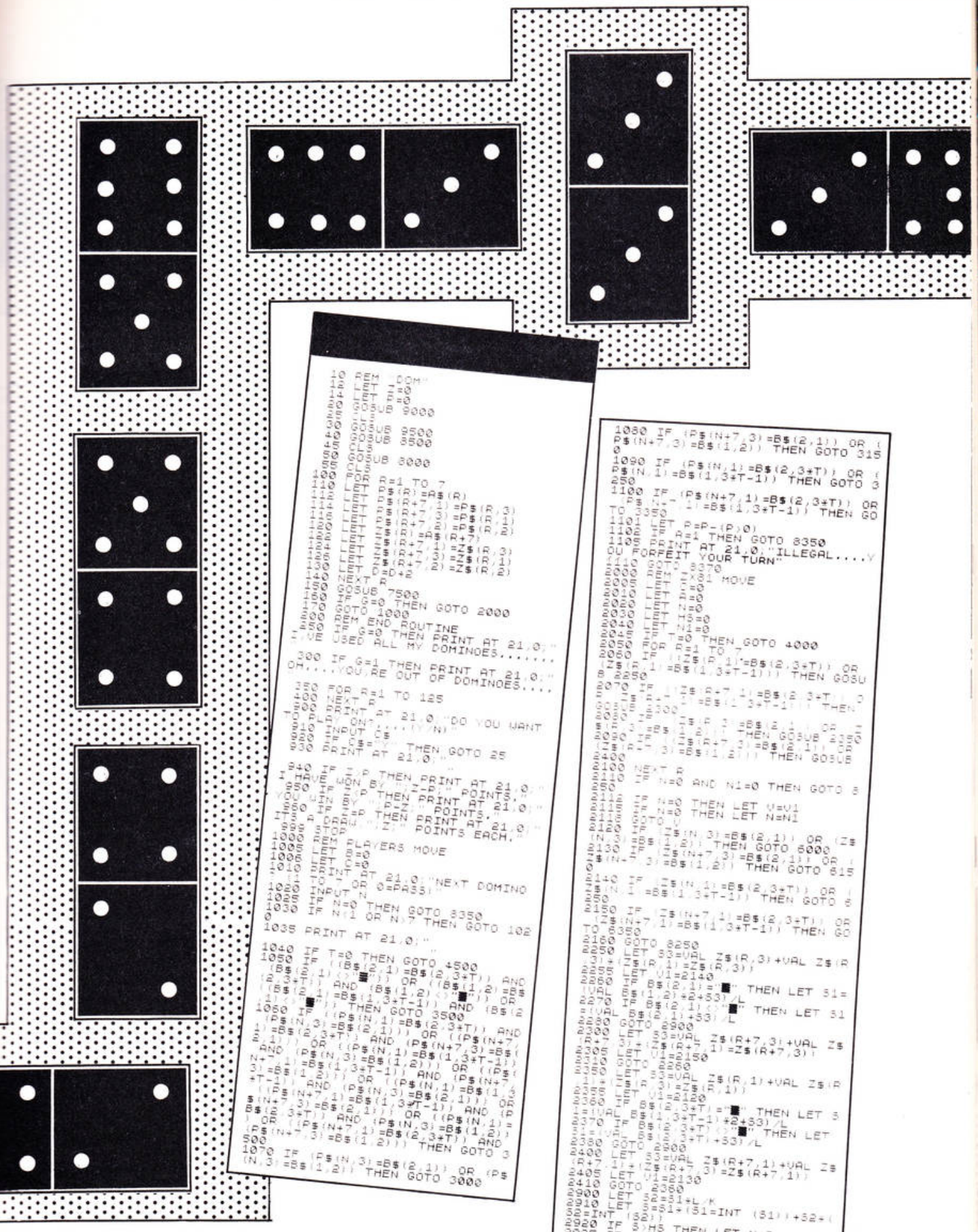


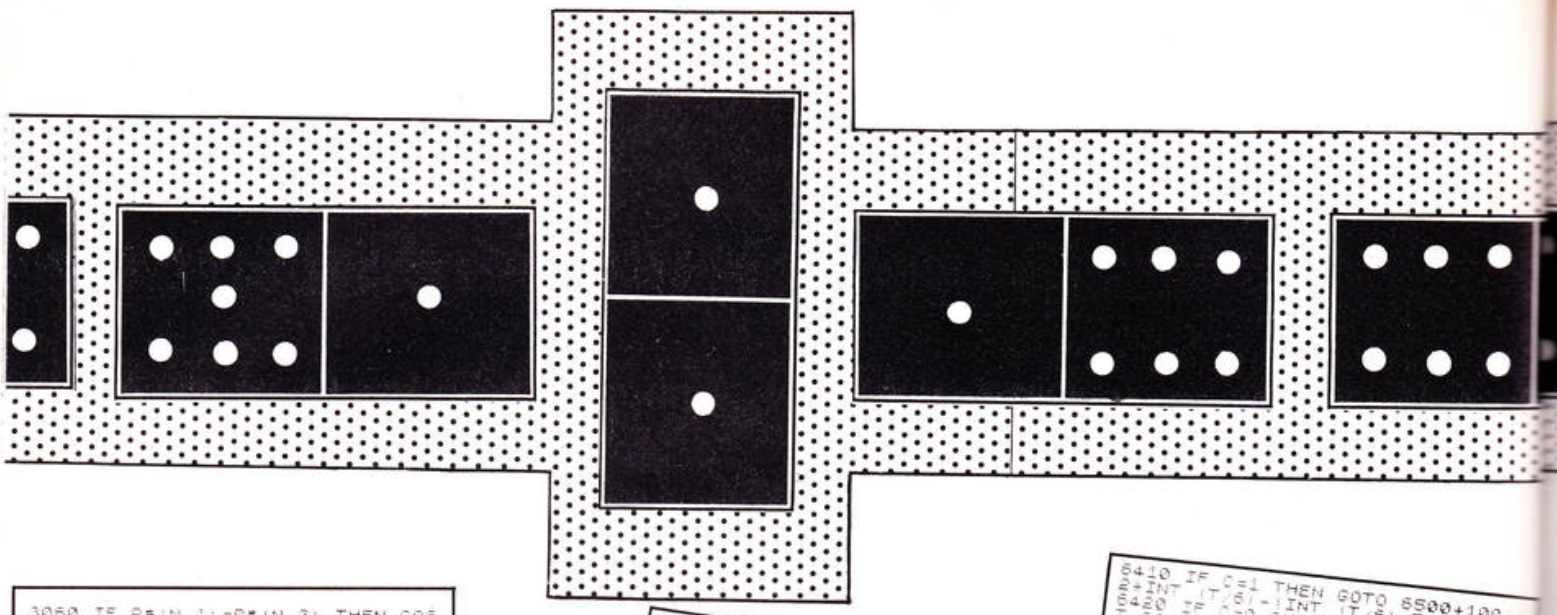
Fig.2. Sample Screen Display





```
10 REM "DOM"
12 LET T=0
14 LET P=0
20 GOSUB 9000
25 CLS
30 GOSUB 9500
40 GOSUB 8500
45 CLS
50 GOSUB 8000
55 CLS
100 FOR R=1 TO 7
110 LET P$(R)=A$(R)
112 LET P$(R+7,1)=P$(R,3)
114 LET P$(R+7,3)=P$(R,1)
116 LET P$(R+7,2)=P$(R,2)
120 LET A$(R)=A$(R+7)
122 LET Z$(R+7,1)=Z$(R,3)
124 LET Z$(R+7,3)=Z$(R,1)
126 LET Z$(R+7,2)=Z$(R,2)
130 NEXT R
150 GOSUB 7500
160 IF G=0 THEN GOTO 2000
170 GOTO 1000
200 REM END ROUTINE
250 IF G=0 THEN PRINT AT 21,0;"
I'VE USED ALL MY DOMINOES......
300 IF G=1 THEN PRINT AT 21,0;"
OH,...YOU RE OUT OF DOMINOES....
350 FOR R=1 TO 125
400 NEXT R
500 PRINT AT 21,0;"DO YOU WANT
TO PLAY ON (Y/N)"
520 INPUT C$
530 IF C$="Y" THEN GOTO 25
540 PRINT AT 21,0;"
"
540 IF T=P THEN PRINT AT 21,0;"
I HAVE WON BY ";Z-P;" POINTS."
550 IF T<P THEN PRINT AT 21,0;"
YOU WIN BY ";P-Z;" POINTS."
560 IF T=P THEN PRINT AT 21,0;"
IT'S A DRAW ";Z;" POINTS EACH."
5900 STOP
6000 REM PLAYERS MOVE
6005 LET B=0
6006 LET C=0
6010 PRINT AT 21,0;"NEXT DOMINO
TO PLAY OR 0=PASS)"
6020 INPUT N
6025 IF N=0 THEN GOTO 8350
6030 IF N<1 OR N>7 THEN GOTO 102
0
6035 PRINT AT 21,0;"
"
6040 IF T=0 THEN GOTO 4500
6050 IF ((B$(2,1)=B$(2,3+T)) AND
(B$(2,1)<>"■")) OR ((B$(1,2)=B$
(2,3+T)) AND (B$(1,2)<>"■")) OR
((B$(2,1)=B$(1,3+T-1)) AND (B$(2
,1)<>"■")) THEN GOTO 3500
1060 IF ((P$(N,1)=B$(2,3+T)) AND
(P$(N,3)=B$(2,1))) OR ((P$(N+7,
1)=B$(2,3+T)) AND (P$(N+7,3)=B$(
2,1))) OR ((P$(N,1)=B$(1,3+T-1))
AND (P$(N,3)=B$(1,2))) OR ((P$(
N+7,1)=B$(1,3+T-1)) AND (P$(N+7,
3)=B$(1,2))) OR ((P$(N,1)=B$(1,3
+T-1)) AND (P$(N,3)=B$(1,3+T-1)
)) AND ((P$(N+7,1)=B$(2,1)) OR
(P$(N+7,3)=B$(2,1))) OR ((P$(N,1)=
B$(2,3+T)) AND (P$(N,3)=B$(1,2)
)) OR ((P$(N+7,1)=B$(2,3+T)) AND
(P$(N+7,3)=B$(1,2))) THEN GOTO 3
500
1070 IF (P$(N,3)=B$(2,1)) OR (P$
(N,3)=B$(1,2)) THEN GOTO 3000
1080 IF (P$(N+7,3)=B$(2,1)) OR (
P$(N+7,3)=B$(1,2)) THEN GOTO 315
0
1090 IF (P$(N,1)=B$(2,3+T)) OR (
P$(N,1)=B$(1,3+T-1)) THEN GOTO 3
250
1100 IF (P$(N+7,1)=B$(2,3+T)) OR
(P$(N+7,1)=B$(1,3+T-1)) THEN GO
TO 3350
1101 LET P=P-(P)0)
1102 IF A=1 THEN GOTO 8350
1105 PRINT AT 21,0;"ILLEGAL....Y
OU FORFEIT YOUR TURN"
1110 GOTO 8370
8000 REM "8X81 MOVE
8005 LET I=0
8010 LET A=0
8020 LET N=0
8030 LET H3=0
8040 LET N1=0
8045 IF T=0 THEN GOTO 4000
8050 FOR R=1 TO 7
8060 IF ((Z$(R,1)=B$(2,3+T)) OR
(Z$(R,1)=B$(1,3+T-1))) THEN GOSU
B 8250
8070 IF ((Z$(R+7,1)=B$(2,3+T)) O
R (Z$(R+7,1)=B$(1,3+T-1))) THEN
GOSUB 8300
8080 IF ((Z$(R,3)=B$(2,1)) OR (Z
$(R,3)=B$(1,2))) THEN GOSUB 8350
8090 IF ((Z$(R+7,3)=B$(2,1)) OR
(Z$(R+7,3)=B$(1,2))) THEN GOSUB
8400
8100 NEXT R
8110 IF N=0 AND N1=0 THEN GOTO 8
8115 IF N=0 THEN LET V=V1
8116 IF N=0 THEN LET N=N1
8117 GOTO V
8120 IF ((Z$(N,3)=B$(2,1)) OR (Z$
(N,3)=B$(1,2))) THEN GOTO 6000
8130 IF ((Z$(N+7,3)=B$(2,1)) OR (
Z$(N+7,3)=B$(1,2))) THEN GOTO 615
0
8140 IF ((Z$(N,1)=B$(2,3+T)) OR (
Z$(N,1)=B$(1,3+T-1))) THEN GOTO 6
150
8150 IF ((Z$(N+7,1)=B$(2,3+T)) OR
(Z$(N+7,1)=B$(1,3+T-1))) THEN GO
TO 6350
8160 GOTO 8250
8250 LET S3=VAL Z$(R,3)+VAL Z$(R
,1)+(Z$(R,1)=Z$(R,3))
8255 LET V1=2140
8260 IF B$(2,1)="■" THEN LET S1=
(VAL B$(1,2)+2+53)/L
8270 IF B$(2,1)<>"■" THEN LET S1
=(VAL B$(2,1)+53)/L
8280 GOTO 8900
8300 LET S3=VAL Z$(R+7,3)+VAL Z$
(R+7,1)+(Z$(R+7,1)=Z$(R+7,3))
8305 LET V1=2150
8310 GOTO 8260
8350 LET S3=VAL Z$(R,1)+VAL Z$(R
,3)+(Z$(R,3)=Z$(R,1))
8355 LET V1=2120
8360 IF B$(2,3+T)="■" THEN LET S
1=(VAL B$(1,3+T-1)+2+53)/L
8370 IF B$(2,3+T)<>"■" THEN LET
S1=(VAL B$(2,3+T)+53)/L
8380 GOTO 8900
8400 LET S3=VAL Z$(R+7,1)+VAL Z$
(R+7,1)+(Z$(R+7,3)=Z$(R+7,1))
8405 LET V1=2130
8410 GOTO 8360
8900 LET S2=S1+L/K
8910 LET S2=S1+(S1=INT (S1))+32+(
S2=INT (S2))
8920 IF S2>H3 THEN LET N=R
8925 IF S2>H3 THEN LET V=V1
8930 IF S2>H3 THEN LET H3=S2
8940 LET N1=R
8950 RETURN
9000 REM "SET UP NEW B$
9010 FOR R=1 TO 3
9020 LET B$(R,4 TO 3+(T+1))=B$(R
,1 TO 3+T)
9030 LET B$(R,1 TO 3)="■■■"
9040 NEXT R
9050 IF P$(N,1)<>P$(N,3) THEN LE
T B$(2,1 TO 3)=P$(N)
```

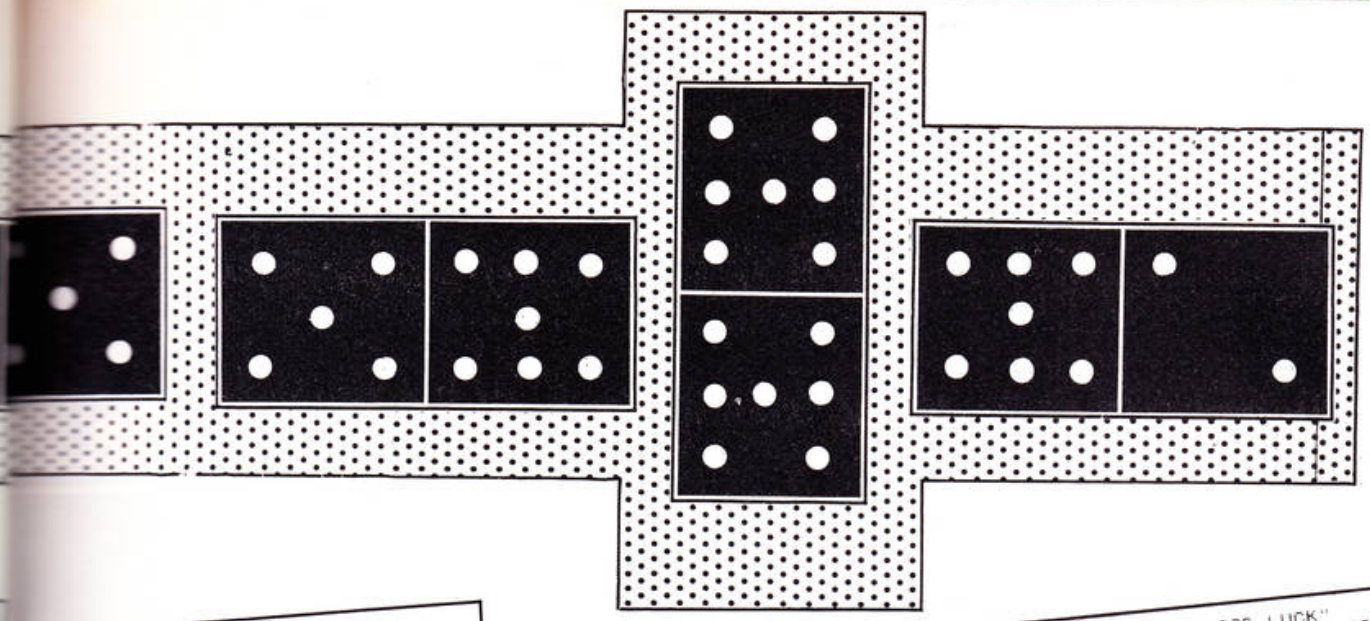
DOMINOES



```
3060 IF P$(N,1)=P$(N,3) THEN GOS
UB 3100
3070 GOTO 6400
3100 FOR R=1 TO 3
3110 LET B$(R,2)=P$(N,R)
3120 NEXT R
3130 RETURN
3150 LET N=N+7
3160 GOTO 3000
3250 IF P$(N,1)<>P$(N,3) THEN LE
T B$(3,3+T+1 TO 3+T+3)=P$(N)
3252 LET C=1
3260 IF P$(N,1)=P$(N,3) THEN GOS
UB 3300
3270 GOTO 6400
3300 FOR R=1 TO 3
3310 LET B$(R,3+(T+1)-1)=P$(N,R)
3320 NEXT R
3330 RETURN
3350 LET N=N+7
3360 GOTO 3250
3500 REM CHOICE OF ENDS
3510 PRINT AT 21,0;"WHICH END?
  (BOTTOM LEFT? Y/N)"
3520 INPUT O$
3525 PRINT AT 21,0;"
3530 IF O$="Y" THEN GOTO 1070
3540 IF O$="N" THEN GOTO 1090
3550 GOTO 3500
4000 REM ZX81 FIRST DOMINO ROUTI
NE
4010 FOR R=1 TO 7
4020 LET S1=VAL Z$(R,1)
4030 LET S2=VAL Z$(R,3)
4040 LET S3=S1+S2
4050 LET S4=S3/L
4060 LET S5=S3/K
4070 LET S=S4+(S4=INT (S4))+S5+(
S5=INT (S5))
4080 IF S>M5 THEN LET N=R
4090 IF S>M5 THEN LET M5=S
4100 NEXT R
4110 IF N=0 THEN LET N=1
4120 IF Z$(N,1)<>Z$(N,3) THEN LE
T B$(3,1 TO 3)=Z$(N)
4130 IF Z$(N,1)=Z$(N,3) THEN GOS
UB 4150
4140 GOTO 6400
4150 FOR R=1 TO 3
4160 LET B$(R,2)=Z$(N,R)
4170 NEXT R
4180 RETURN
4500 REM PLAYER 1ST DOMINO ROUTI
NE
4510 IF (P$(N,1)<>P$(N,3)) THEN
LET B$(2,1 TO 3)=P$(N)
4520 IF (P$(N,1)=P$(N,3)) THEN G
OSUB 4950
4540 GOTO 6400
4950 REM ENTER DOUBLES INTO B$
4960 FOR R=1 TO 3
4970 LET B$(R,T+2)=P$(N,R)
4980 NEXT R
4990 RETURN
5000 REM PRINT STARTING DOMINO
5002 FOR R=1 TO 3
5004 IF C=0 THEN PRINT AT 16+R,1
3;B$(R,1 TO 3)
5006 NEXT R
5008 REM TAKE NEW DOMINO ROUTINE
5010 LET C=0+1
5020 LET N=N-7+(N<7)
5030 IF G=1 THEN GOTO 5100
5040 LET P$(N)=" "
5050 LET P$(N+7)=" "
5060 IF C=1 THEN GOSUB 5400
5070 GOTO 5500
5100 LET P$(N)=" "
5110 LET P$(N+7)=" "
5120 IF C=1 THEN GOSUB 5300
5130 GOTO 5500
5300 LET P$(N)=A$(D)
5310 FOR R=1 TO 3
```

```
5320 LET P$(N+7,R)=A$(D,4+R)
5330 NEXT R
5340 PRINT AT 12+N,4;P$(N)
5350 RETURN
5400 LET P$(N)=A$(D)
5410 FOR R=1 TO 3
5420 LET P$(N+7,R)=A$(D,4+R)
5430 NEXT R
5440 RETURN
5500 REM KEEP SCORE ROUTINE
5510 PRINT AT 3+H5,
5520 GOTO 5750
5530 IF B$(2,1)="■" THEN LET S1=
VAL B$(1,2)
5540 IF B$(2,1)="■" THEN LET S1=
VAL B$(1,2)
5550 IF B$(2,3+T)="■" THEN LET S
VAL B$(1,3+T)
5560 IF B$(2,3+T-1)="■" THEN LET
VAL B$(1,3+T-1)
5570 LET S2=S1+S2
5580 LET S3=S2+S2
5590 LET S4=S3/L
5600 LET S5=S3/K
5610 LET S=S4+(S4=INT (S4))+S5+(
S5=INT (S5))
5620 LET P=P+S
5630 PRINT AT 4,4;P
5640 REM WHO GOES NEXT ROUTINE
5650 IF A+B=0 THEN GOTO 900
5660 FOR R=1 TO 7
5670 IF P$(R)=" " THEN GOTO 5
5680 NEXT R
5690 LET P=P+1
5700 PRINT AT 4,4;P
5710 GOTO 800
5720 FOR R=1 TO 7
5730 IF Z$(R)=" " THEN GOTO 5
5740 NEXT R
5750 LET Z=Z+1
5760 PRINT AT 8,4;Z
5770 GOTO 800
5780 G=(G=0)
5790 IF G=0 THEN PRINT AT 21,0;"
MY TURN NOW.....
5990 GOTO 1000+(2-G)
6000 REM PICK UP NEW B$
6010 FOR R=1 TO 3
6020 LET B$(R,4 TO 3+(T+1))=B$(R
,1 TO 3)="■■■"
6030 NEXT R
6040 IF Z$(N,1)<>Z$(N,3) THEN LE
T B$(R,1 TO 3)=Z$(N)
6050 IF Z$(N,1)=Z$(N,3) THEN GOS
UB 6100
6060 GOTO 6400
6100 FOR R=1 TO 3
6110 LET B$(R,2)=Z$(N,R)
6120 NEXT R
6130 RETURN
6150 LET N=N+7
6160 GOTO 6000
6200 REM SHOW DOMINOES ON TABLE
6400 LET T=T+1
6405 IF G=1 THEN PRINT AT 12+N-(
```

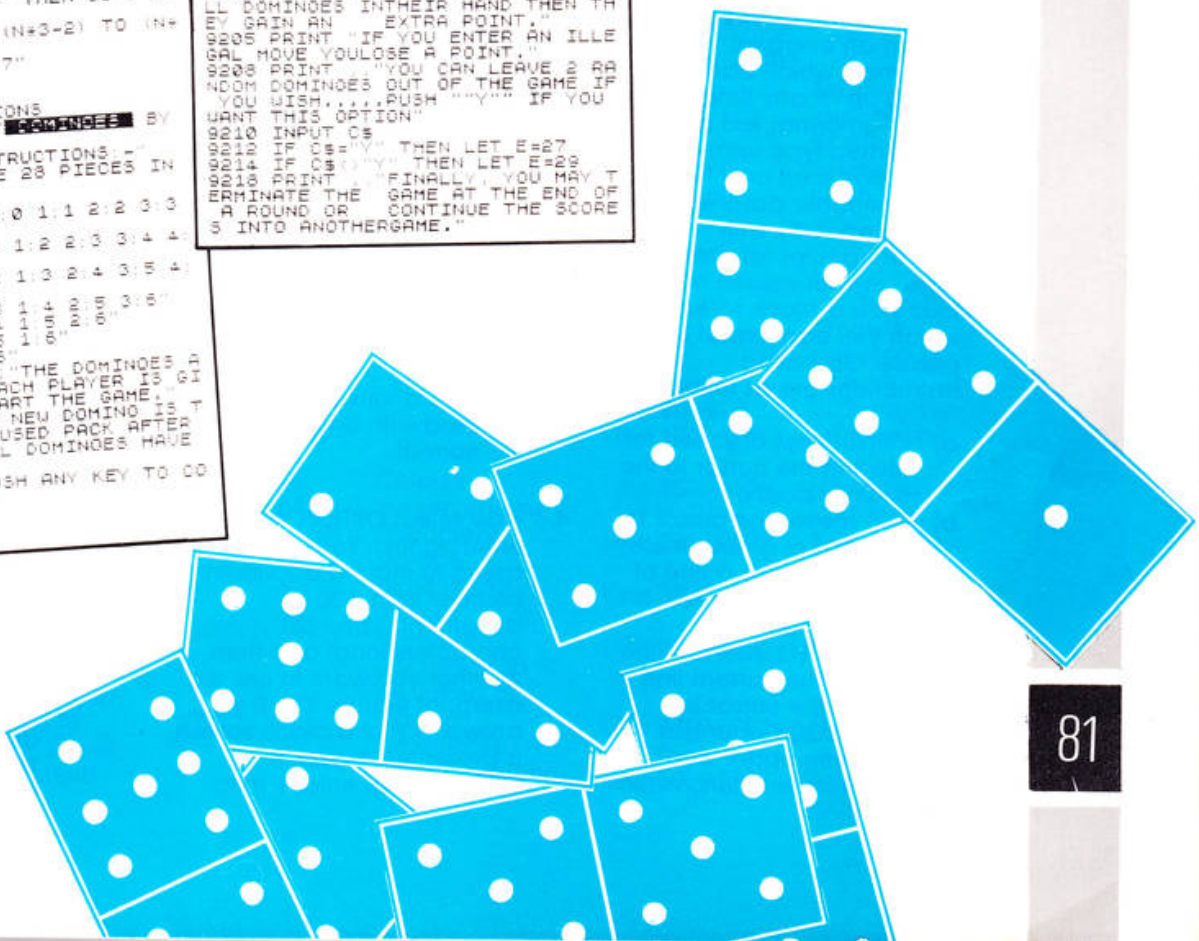
```
6410 IF G=1 THEN GOTO 6500+100-
6420 IF (T/6)=INT (T/6)=T/6)
6430 IF (T/6)=INT (T/6) AND T/5 THEN PRINT A
T 21,0;"I'M JUST MOVING THE PIE
6500 REM DOMINOES 1 TO 5
6510 FOR R=1 TO 3
6520 IF C=1 THEN PRINT AT 16+R,1
6525 IF C=0 THEN PRINT AT 16+R,1
6530 NEXT R
6540 NEXT R
6600 IF T=6 THEN GOTO 5000
6610 REM DOMINO NO. 6
6620 FOR R=1 TO 3
6630 FOR I=1 TO 3
6635 PRINT AT 18-I,26+R;B$(R,15+
I)
6640 NEXT I
6650 NEXT R
6700 IF T=6 THEN GOTO 5000
6705 REM DOMINOES 7 TO 11
6710 FOR R=1 TO 3
6720 FOR I=1 TO 15
6730 LET S$(R,16-I)=B$(R,16+I)
6740 NEXT I
6745 PRINT AT 12+R,13;S$(R)
6750 NEXT R
6760 IF T=12 THEN GOTO 5000
6800 REM DOMINO NO. 12
6810 FOR R=1 TO 3
6820 FOR I=1 TO 3
6830 PRINT AT 14-I,10+R;B$(R,33+
I)
6840 NEXT I
6850 NEXT R
6860 IF T=12 THEN GOTO 5000
6900 REM DOMINOES 13 TO 17
6910 FOR R=1 TO 3
6920 PRINT AT 8+R,13;B$(R,37 TO
)
6930 NEXT R
6940 IF T=18 THEN GOTO 5000
7000 REM DOMINO NO. 18
7010 FOR R=1 TO 3
7035 FOR I=1 TO 3
7038 PRINT AT 10-I,26+R;B$(R,51+
I)
7045 NEXT I
7050 NEXT R
7055 IF T=18 THEN GOTO 5000
7100 REM DOMINOES 19 TO 23
7110 FOR R=1 TO 3
7135 FOR I=1 TO 15
7145 LET S$(R,16-I)=B$(R,54+I)
7150 PRINT AT 4+R,13;S$(R)
7155 NEXT R
7200 IF T=24 THEN GOTO 5000
7210 REM DOMINO NO. 24
7220 FOR R=1 TO 3
7235 FOR I=1 TO 3
7238 PRINT AT 6-I,10+R;B$(R,69+I
)
7245 NEXT I
7250 NEXT R
7300 IF T=24 THEN GOTO 5000
7310 REM DOMINOES 25 TO 27
7325 FOR R=1 TO 3
7335 PRINT AT R,13;B$(R,73 TO )
7340 GOTO R
7500 REM DRAW BOARD
7510 PRINT "
7515 FOR R=1 TO 19
7520 PRINT "
7530 NEXT R
7540 PRINT "
7550 PRINT AT 11,0;"
7560 PRINT AT 12,0;"
7570 PRINT AT 13,0;"
7580 PRINT AT 6,23;"
7585 PRINT AT 6,2;"PLAYER"
7590 PRINT AT 4,4;P
7595 PRINT AT 6,3;"ZX81"
7600 FOR R=1 TO 7
7610 PRINT AT 12+R,2;R
7620 NEXT R
7630 FOR R=1 TO 7
7640 PRINT AT 12+R,4;P$(R)
7650 NEXT R
7990 RETURN
8000 REM WHO GOES FIRST ROUTINE
8010 PRINT "TO DECIDE WHO GOES F
```

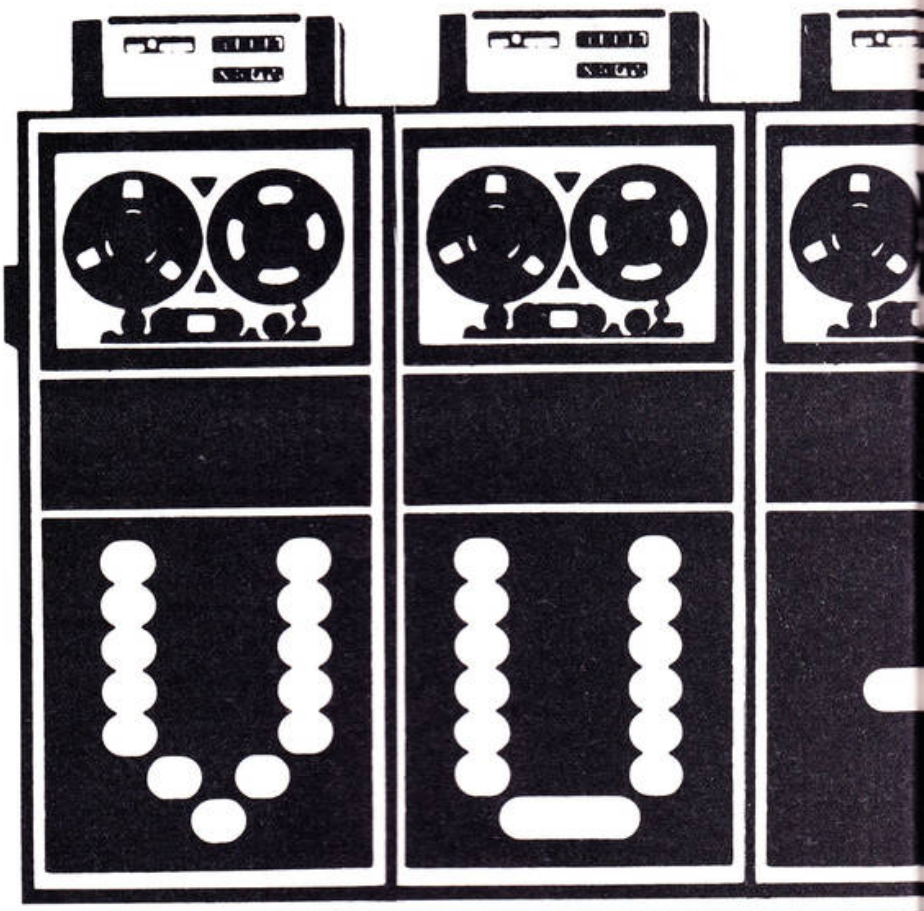


```
WILL THROW A DICE...THE H
THROWCAN START."
CAUSE 200
PRINT ;;"PLAYERS DICE = ";
LET D1=INT (RND*6)+1
PRINT D1
CAUSE 200
PRINT ;;"    ZX81 DICE = ";
LET D2=INT (RND*6)+1
IF D2=D1 THEN GOTO 8060
PRINT D2
CAUSE 200
IF D1>D2 THEN PRINT ;;"OK..
GO FIRST."
IF D1>D2 THEN LET G=1
IF D2>D1 THEN PRINT ;;"OK..
WILL START."
PRINT ;;"PUSH ANY KEY TO CO
NTINUE.....";
PAUSE 40000
RETURN
REM ZX81 CANT GO ROUTINE
PRINT AT 21,0;"OH DEAR I CA
N GO.....";
LET A=1
FOR R=1 TO 50
NEXT R
GOTO 5750
REM PLAYER CANT GO ROUTINE
PRINT AT 21,0;"OH...YOU CAN
'T.....";
LET B=1
FOR R=1 TO 50
NEXT R
GOTO 5750
REM SHUFFLE DOMINOES
PRINT ;;"I'M JUST SHUFFLING
DOMINOES."
RAND
DIM A$(28,3)
FOR R=1 TO 28
LET N=INT (RND*28)+1
IF D$(N*3)="7" THEN GOTO 85
70
LET A$(R)=D$(N*3-2) TO (N*
3)
LET D$(N*3)="7"
NEXT R
RETURN
REM INSTRUCTIONS
PRINT TAB 5;" DOMINOES BY
N.BROOKS."
PRINT ;;"INSTRUCTIONS:-"
PRINT ;;" THE 28 PIECES IN
THE SET ARE:-"
PRINT ;;"  0:0 1:1 2:2 3:3
4:4 5:5 6:6"
PRINT "  0:1 1:2 2:3 3:4 4:
5 5:6"
PRINT "  0:2 1:3 2:4 3:5 4:
6"
PRINT "  0:3 1:4 2:5 3:6"
PRINT "  0:4 1:5 2:6"
PRINT "  0:5 1:6"
PRINT "  0:6"
PRINT TAB 7;"THE DOMINOES A
RE THEN SHUFFLEDAND EACH PLAYER IS GI
VEN SEVEN  TO START THE GAME."
PRINT ;;" A NEW DOMINO IS T
AKEN FROM THE UNUSED PACK AFTER
EACH GO UNTIL ALL DOMINOES HAVE
BEEN USED."
PRINT ;;"PUSH ANY KEY TO CO
NTINUE.....";
PAUSE 4E4
CLS
```

```
9046 PRINT TAB 5;" DOMINOES BY
N.BROOKS."
9048 PRINT ;;"YOU AND THE ZX81 L
AY ALTERNATELY"
9050 PRINT ;;" A DOMINO CAN ONLY
 BE USED IF ITHAS A NUMBER MATCH
ING AN END NO.ON THE TABLE."
9060 PRINT "DOUBLES (EG 3:3)  AR
E LAID CROSSWISE AND BOTH NOS.AR
E USED IN SCORING."
9070 PRINT ;;"SCORING:-"
9080 PRINT ;;" TAB 10;"ADD THE END
 NUMBERS ONTHE TABLE AND DIVIDE
BY 5 OR 3  OR BOTH."
9090 PRINT "(EG 4:33:5 SCORES 3
PTS)"
9095 PRINT ;;"YOU CAN SCORE USIN
G 6 AND 4 IF  YOU PREFER....."
9096 PRINT ;;"ENTER OPTION REQD:
- ""A"" = 5 AND 3"
9097 PRINT TAB 20;"""B"" = 6 AND
 4"
9100 INPUT C$
9102 IF C$="A" THEN LET K=5
9104 IF C$="A" THEN LET L=3
9106 IF C$="B" THEN LET K=6
9108 IF C$="B" THEN LET L=4
9110 IF C$<>"A" AND C$<>"B" THEN
 GOTO 9100
9115 CLS
9120 PRINT TAB 5;" DOMINOES BY
N.BROOKS."
9130 PRINT ;;"YOU HAVE CHOSEN TH
E ";K;"/";L;" METHOD  OF SCORING
."
9200 PRINT ;;"IF A PLAYER USES A
LL DOMINOES INTHEIR HAND THEN TH
EY GAIN AN  EXTRA POINT."
9205 PRINT "IF YOU ENTER AN ILLE
GAL MOVE YOULOSE A POINT."
9208 PRINT "YOU CAN LEAVE 2 RA
NDOM DOMINOES OUT OF THE GAME IF
 YOU WISH.....PUSH ""Y"" IF YOU
WANT THIS OPTION"
9210 INPUT C$
9212 IF C$="Y" THEN LET E=27
9214 IF C$<>"Y" THEN LET E=29
9218 PRINT ;;"FINALLY YOU MAY T
ERMINATE THE  GAME AT THE END OF
 A ROUND OR  CONTINUE THE SCORE
S INTO ANOTHERGAME."
```

```
9220 PRINT TAB 11;"GOOD LUCK"
9450 PRINT ;;"PUSH ANY KEY TO CO
NTINUE.....";
9460 PAUSE 4E4
9490 RETURN
9500 REM SET UP VARIABLES
9505 PRINT "PLEASE WAIT.....";
9530 LET D$="0:0 0:1 0:2 0:3 0:4 0:5 0:6
1:1 1:2 1:3 1:4 1:5 1:6 2:2 2:3 2:4 2:5 2:6
3:3 3:4 3:5 3:6 4:4 4:5 4:6 5:5 5:6 6:6"
9540 LET G=0
9550 DIM T$(14,3)
9560 DIM P$(14,3)
9570 DIM B$(3,81)
9580 LET R$=" "
9590 FOR R=1 TO 27
9600 LET R$=R$+" "
9610 NEXT R
9620 FOR R=1 TO 3
9630 LET B$(R)=R$
9640 NEXT R
9650 LET D=0
9660 LET T=0
9670 LET A=0
9680 LET B=0
9690 LET V=2120
9700 LET V1=0
9710 LET C=0
9720 DIM S$(3,15)
9990 RETURN
9995 REM MEMORY USED
9997 PRINT PEEK 16396+256*PEEK 1
6397-16509
9999 STOP
```





BY ALASTAIR DE WET

This general document storage/word processing home utility program is written in BASIC for the 48K Spectrum

The program on loading will present you with a main menu screen with 9 options displayed. These can be used in any sequence, although it will be obvious that certain of the options — eg, hard copy printing of enquiries — depend on a certain amount of text having already been entered. A logical sequence of operation would be as follows:

1. **OPTION 1** — Select Colours: You will be asked to input ink, paper, border, and brightness options. These options will then be used when inputing or editing your text. This option can be selected at any time to change the colours. If this option is not called, colours will be pre-set to Paper 7, Ink 0, Border 3, Bright 1. Once you have entered your options the program returns to the main menu.

2. **OPTION 3** may be selected next. This option gives you the facility to change to a different character set. If you select 'long-hand' the new set of characters is poked into memory. This takes half a minute or so the first time you enter this routine, but on calling this facility again you will find that the changeover from normal to long hand is almost instantaneous.

3. **OPTION 7** should now be selected. This allows you to design an overprint document which can be called up at any time when entering normal text as a pre-printed type form on to which you then only have to enter variable data. You are presented firstly with operating rules for entering data on your screen and then with a blank screen which you can format as you please. Operating instructions are as follows:

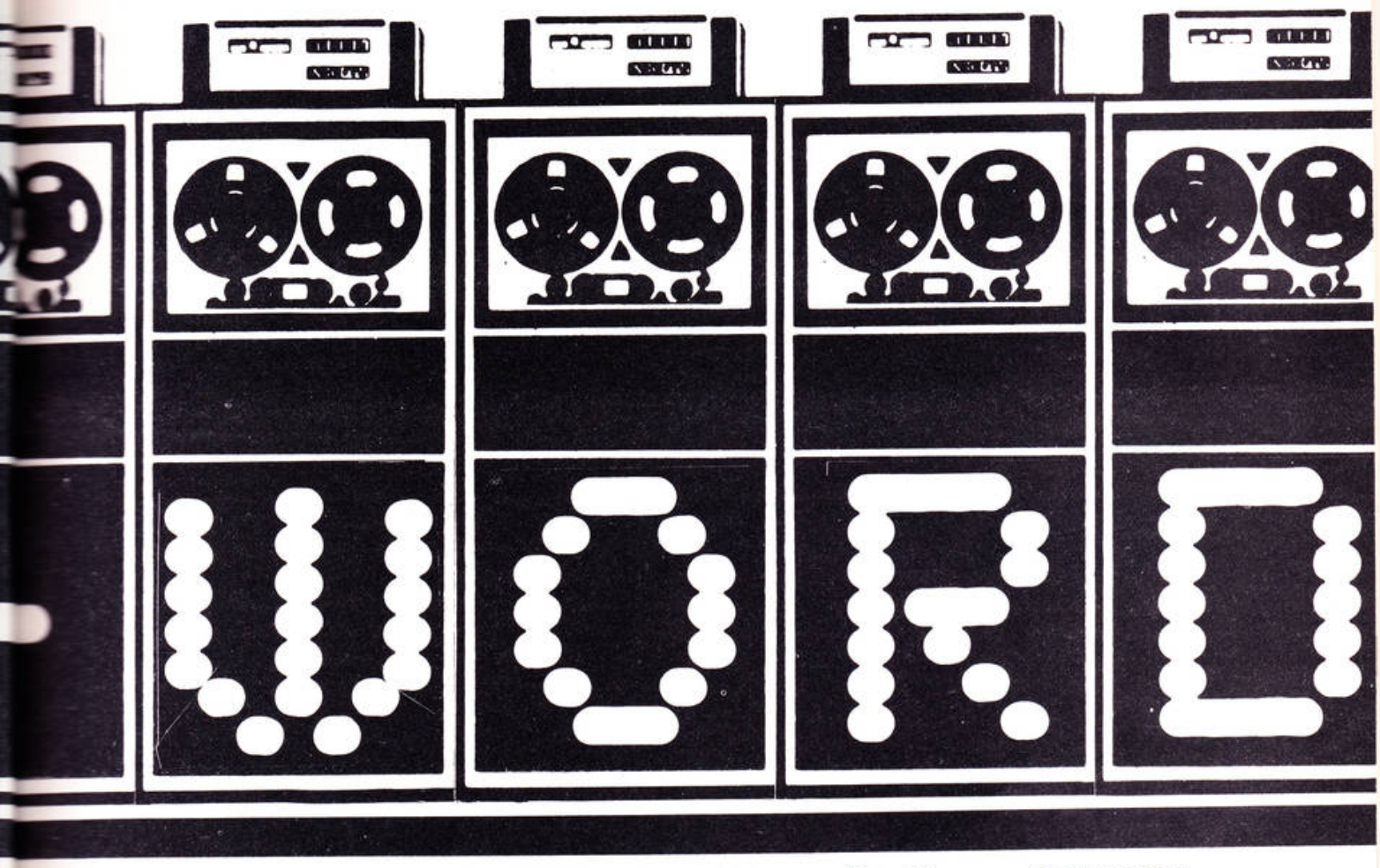
   a) Cursor keys are used for moving the cursor to the desired position.
   b) EDIT moves the cursor to the start of a new line.
   c) DELETE deletes a line of text where the cursor is positioned.
   d) TRUE VIDEO inserts a line before the current line where the cursor is.
   e) CAPS LOCK changes between upper and lower case letters. When depressing, a high tone beep indicates upper case and a low tone lower case.
   f) INVERSE VIDEO quits and returns you to the main menu without storing any of the entries currently being made to the document on which you were working.
   g) ENTER is pressed once you have completed everything you want to do. The document is then stored away for later recall.
   h) Operations a) thru f) are used with caps shift as normal.

4. Now select **OPTION 4** for entering text. You will be asked to input a document number from 1-30, a document name (20 characters long) and then whether you want to use the overprint feature Y/N. If you answer N (no) a blank screen is presented to you. If, however, you enter Y then your pre-printed document is flashed on to the screen. You now enter your text in free format using the rules as described in 3(a)-3(h). The entire screen image is saved when you press ENTER and you return to the main menu. You can now carry on entering more documents selecting OPTION 4.

5. **OPTION 8** give you the facility to print a copy of any of your documents to your printer. Enter the relevant document number and the print begins and returns to the main menu of completion.

6. Entering **OPTION 2** (Edit/Enquiries) presents you with a sub-menu with further options available:

   a) PRINT CATALOGUE — Displays all the current documents 1-30 and the name of each as a quick reference guide to help you retrieve the document you need.



b) RETRIEVE DOCUMENT — Here you can call up any of your previously entered documents on to the screen for reference purposes or for altering. When altering a document follow the same rules as in 3(a) — 3(h) changing the image as displayed on the screen and press ENTER when you want it stored. Remember that you can only retrieve documents that already exist in your catalogue file.

c) DELETE DOCUMENT removes all reference to the document from the system and frees the document for later use.

d) CHANGE DOCUMENT NAME allows you to alter the document name.

e) TRANSFER DOCUMENT. If you wish the document entered as, say No.12, to now become No.1, you can use this option to do just that. This option does not delete the original document from position 12, and you will have to do this if you need to using Option 6(c) above.

f) RETURN TO MAIN MENU does just that.

7. **OPTION 5** — Save Data: Is used before you switch off to secure all your documents on to tape to be loaded back by OPTION 6 — Load Data, when you wish to access the data again.

8. **OPTION 9** — End Job: Will stop the program after making sure you have saved your data. If you have not, it returns to the main menu to allow you the chance to do so.

WHAT IF I ACCIDENTALLY BREAK INTO OR CRASH PROGRAMS?
DON'T TYPE: RUN
TYPE: GO TO 10 and your data will not be lost.

Variables

DIM A(1-4)	Colours
DIM T$ (1,640)	Screen image
DIM E$ (31,660)	Document storage
A$	Main menu options
S$	Name of document to be saved or loaded
POS	Calculated current position in T$
X and Y	Cursor position, row and column
D	Document no.
O	Sub menu Options
CH,A,B,C,F,I$,0$	General variables

Line By Line

Line 1-6	Set up values in variables, poke UDG's and keyboard click.
Line 10-200	Main program logic displays main menu and directs program to the process selected by operator.
Line 800-900	UDG routine.
Line 1000-1250	Main menu display.
Line 1500-1600	Hard copy print routine.
Line 2000-2500	Select colours routine.
Line 3000-3200	Sub menu print routine.
Line 3500-3997	Routines performed from sub menu.
Line 4000-4900	New char set routine.
Line 5000-5100	Save data routine.

Line 5500-5700	Load data routine.
Line 6000-6200	End program routine.
Line 6500-6800	Overprint document format routine.
Line 7000-7450	Enter text.
Line 7500-7590	Store text.
Line 7600-7690	Delete and insert line subroutine.
Line 7700-7790	Instructions.
Line 7800-7850	Cursor movement.
Line 9000	Displays program end address.
Line 9700-9750	Sets up DIM size to 31×660.

Notes

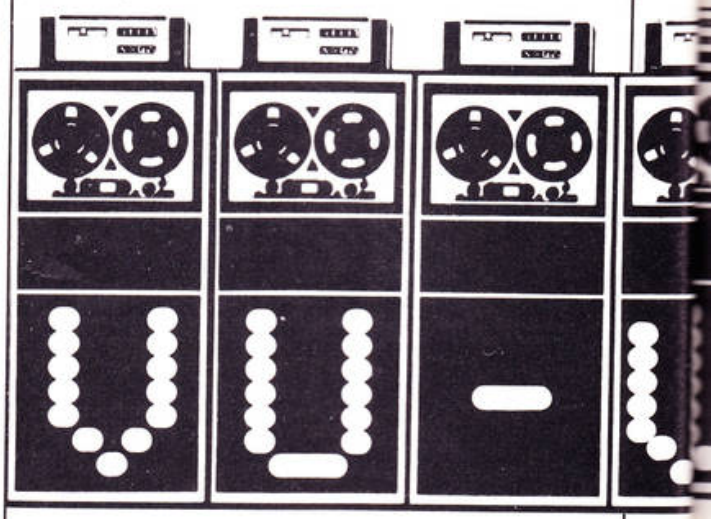
1. A few thousand bytes of available memory have not been utilized as they could come in very useful in future modifications to the program. These enhancements could include deleting and inserting at WORD rather than LINE level with automatic wrap around taking place as in professional word processing packages. This feature, however, may be a little slow in BASIC.
2. The program has been saved with its dimension E$ size reduced to 1 by 660. This automatically reverts back to 31 by 660 by lines 9700 through 9750, which saves a lot of loading time. To do this, change line 9710 to DIMX=1, Type RUN, Break into program, change line 9710 back to 31, (Do not type RUN), type SAVE "VU-WORD" LINE 1. The program is saved with its size reduced by 30×660 bytes.
3. Other possible enhancement ideas could include:
   a) A bubble sort on a given part of each ducument
   b) An enquiry facility at string level to retrieve all documents with a given string appearing anywhere within the text.

```
   1 DIM a(4): LET A(1)=7: LET A
(4)=1: LET A(3)=3: GO SUB 9700
   2 DIM E$(DIMX,DIMY): DIM T$(1
,640)
   3 LET C=0: LET CH=0: LET D=0
   4 RESTORE 840
   5 GO SUB 800
   6 POKE 23609,100
  10 GO SUB 1000
  15 BEEP .5,30
  20 IF A$="1" THEN  GO SUB 2000
  30 IF A$="2" THEN  GO SUB 3000
  40 IF A$="3" THEN  GO SUB 4000
  50 IF A$="4" THEN  GO SUB 7000
 100 IF A$="5" THEN  GO SUB 5000
 120 IF A$="6" THEN  GO SUB 5500
 130 IF A$="7" THEN  GO SUB 6500
 140 IF A$="8" THEN  GO SUB 1500
 150 IF A$="9" THEN  GO SUB 6000
 200 GO TO 10
 800 FOR A=0 TO 7
 810 READ B
 820 POKE USR "A"+A,B
 830 NEXT A
 840 DATA 85,170,85,170,85,170,8
5,170
 900 RETURN
1000 REM main menu
1010 PAPER 7: BRIGHT 1: INK 0: B
ORDER 7: CLS
1012 FOR A=0 TO 21
1015 PRINT AT A,0; INK 7; PAPER
6;"@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@"
1017 NEXT A
1020 PRINT AT 0,4; PAPER 6; INK
0;"** VU-WORD **  A DE WET"
1030 PRINT AT 1,4; PAPER 6; INK
0;"------------------------"
1040 PRINT AT 2,10; PAPER 6; INK
 0;"MAIN MENU"
1050 PRINT AT 3,10; PAPER 6; INK
 0;"**********"
1060 PRINT AT 6,3; PAPER 6; INK
0;"1  -  SELECT COLOURS"
1070 PRINT AT 7,3; PAPER 6; INK
0;"2  -  EDIT/ENQUIRE FACILITY"
1080 PRINT AT 8,3; PAPER 6; INK
0;"3  -  SELECT CHARACTER SET"
1090 PRINT AT 9,3; PAPER 6; INK
0;"4  -  ENTER TEXT"
1100 PRINT AT 10,3; PAPER 6; INK
 0;"5  -  SAVE DATA"
1110 PRINT AT 11,3; PAPER 6; INK
 0;"6     LOAD DATA"
1115 PRINT AT 12,3; PAPER 6; INK
 0;"7  -  FORMAT OVERPRINT DOC."
1118 PRINT AT 13,3; PAPER 6; INK
 0;"8  -  HARD COPY PRINT"
1120 PRINT AT 14,3; PAPER 6; INK
 0;"9  -  END JOB"
1150 INPUT "ENTER OPTION REQUIRE
D ";A$
1250 RETURN
1500 REM H.C.P.
1510 GO SUB 3600
1520 IF D>30 THEN  GO TO 1600
1530 PRINT AT 0,0; INVERSE 1;"DO
C ";D;"  ";E$(D,641 TO 660)
1540 PRINT AT 1,0;T$(1,1 TO 640)
1550 COPY
1590 LET T$(1)="  "
1595 BEEP .5,0
1600 RETURN
2000 REM SET COLOURS
2005 PAPER 7: CLS
2007 PRINT AT 0,5; INVERSE 1;" S
ELECT COLOURS "
2010 FOR A=1 TO 100
2020 POKE 22560+INT (RND*672),IN
T (RND*127)
2030 NEXT A
2040 INPUT "ENTER PAPER COLOUR "
;A(1)
```

```
2050 INPUT "ENTER  INK  COLOUR "
;A(2)
2060 INPUT "ENTER BORDER COLOUR
";A(3)
2070 INPUT "ENTER BRIGHT 1 OR 0
";A(4)
2080 PAPER A(1): INK A(2): BORDE
R A(3): BRIGHT A(4): CLS
2500 RETURN
3000 REM EDIT/ENQUIRY
3010 CLS
3020 PRINT AT 0,5; INVERSE 1;" S
UB MENU ENQUIRY/EDIT "
3030 PRINT AT 1,5;"-------------
--------"
3040 PRINT AT 5,3;"1  -  PRINT C
ATALOGE"
3050 PRINT AT 6,3;"2  -  RETRIEV
E DOCUMENT "
3060 PRINT AT 7,3;"3  -  DELETE
ENTIRE DOCUMENT"
3070 PRINT AT 8,3;"4  -  CHANGE
DOCUMENT NAME"
3080 PRINT AT 9,3;"5  -  TRANSFE
R DOCUMENT"
3090 PRINT AT 10,3;"6  -  RETURN
 TO MAIN MENU"
3100 INPUT "SELECT REQUIRED OPTI
ON ";O
3110 IF O=1 THEN  GO SUB 3500: G
O TO 3000
3120 IF O=2 THEN  GO SUB 3600
3125 IF O=2 AND D<31 THEN  GO SU
B 7782: GO TO 7020
3130 IF O=3 THEN  GO SUB 3700: G
O TO 3000
3140 IF O=4 THEN  GO SUB 3800: G
O TO 3000
3150 IF O=5 THEN  GO SUB 3900: G
O TO 3000
3160 IF O=6 THEN  GO TO 3200
3180 IF O>6 THEN  GO TO 3100
3190 GO TO 3000
3200 RETURN
3500 REM PRINT CAT
3510 CLS
3520 PRINT AT 0,5; INVERSE 1;" C
ATALOGE FILE "
3540 FOR A=1 TO 20
3545 PRINT AT A,3;A;"   ";E$(A,6
41 TO 660)
3550 NEXT A
3560 PRINT AT 21,0;"PRESS ENTER
TO CONTINUE"
3565 PAUSE 0
3570 FOR A=21 TO 31
3575 PRINT AT A-20,3;A;"   ";E$(
A,641 TO 660)
3580 NEXT A
3582 PRINT AT 12,0;,,,,,,,,,,,,,
,,,,,
3585 PAUSE 0
3590 RETURN
3600 REM RETRIEVE
3605 CLS
3610 PRINT AT 0,5; INVERSE 1;" D
OCUMENT RETRIEVAL "
3615 IF A$="8" THEN  PRINT AT 0,
5; INVERSE 1;"  HARD COPY PRINT
  "
3620 PRINT AT 10,1;"FOLLOW THE I
NSTRUCTIONS BELOW"
3630 INPUT "ENTER DOCUMENT NO 1-
30 ";D
3635 IF D>30 THEN  GO TO 3690
3640 IF E$(D,641 TO 645)="      "
 THEN  GO SUB 3995: GO TO 3630
3660 LET T$(1)=E$(D)
3690 RETURN
3700 REM DELETE
3705 CLS
3710 PRINT AT 0,5; INVERSE 1;" D
OCUMENT DELETION "
3720 PRINT AT 10,1;"FOLLOW THE I
NSTRUCTIONS BELOW"
3730 INPUT "ENTER DOCUMENT NO 1-
30 ";D
3735 IF D>30 THEN  GO TO 3790
3740 IF E$(D,641 TO 645)="      "
 THEN  GO SUB 3995: GO TO 3730
3750 PRINT AT 12,3; FLASH 1;"WAI
T WHILE I DELETE";AT 13,3; FLASH
 1;" DOCUMENT NO. ";D
3760 LET E$(D)=" "
3770 BEEP 5,1
3790 RETURN
3800 REM CHANGE DOCUMENT NAME
3805 CLS
3810 PRINT AT 0,5; INVERSE 1;" D
OCUMENT NAME CHANGE "
3820 PRINT AT 10,1;"FOLLOW THE I
NSTRUCTIONS BELOW"
3830 INPUT "ENTER DOCUMENT NO 1-
30  ";D
3835 IF D>30 THEN  GO TO 3890
3840 IF E$(D,641 TO 645)="      "
 THEN  GO SUB 3995: GO TO 3830
3850 PRINT AT 12,0;"CURRENT NAME
 OF DOCUMENT IS :": PRINT AT 13,
0;"   ";E$(D,641 TO 660)
3860 INPUT "NEW DOCUMENT NAME IS
 ? ";E$(D,641 TO 660)
3870 PRINT AT 15,0; FLASH 1;"NEW
 NAME OF DOCUMENT IS :": PRINT A
T 16,0; FLASH 1;"   ";E$(D,641 T
O 660)
3880 BEEP 5,1
3890 RETURN
3900 REM TRANSFER DOCUMENT
3905 CLS
3910 PRINT AT 0,5; INVERSE 1;" D
OCUMENT TRANSFER "
```

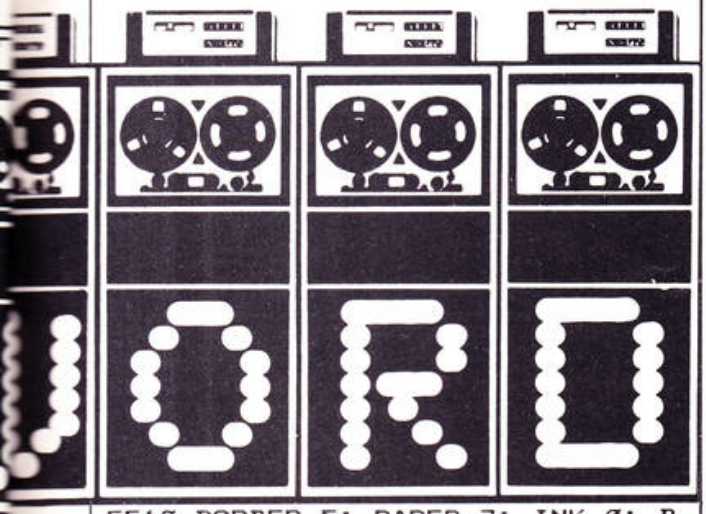
```
3920 PRINT AT 10,1;"FOLLOW THE I
NSTRUCTIONS BELOW"
3930 INPUT "ENTER DOC TO BE TRAN
SFERED  ";D
3935 IF D>30 THEN  GO TO 3990
3940 IF E$(D,641 TO 645)="      "
 THEN  GO SUB 3995: GO TO 3930
3960 PRINT AT 12,0;"DOC NO.     D
OCUMENT NAME     " : PRINT AT 13,
0;"    ";D;"          ";E$(D,641 TO
660)
3965 INPUT "NEW NO. OF DOCUMENT
IS ? ";E
3966 IF E>30 THEN  GO TO 3990
3967 IF E$(D,641 TO 645)="      "
 THEN  GO SUB 3995: GO TO 3965
3970 PRINT AT 15,0; FLASH 1;"DOC
UMENT ";D;" HAS BEEN TRNSFRD": P
RINT AT 16,0; FLASH 1;"TO NEW PO
SITION AT NO ";E: PRINT AT 17,0;
 FLASH 1;"OLD DOC HAS NOT BEEN D
ELETED"
3975 LET E$(E)=E$(D)
3985 BEEP 5,1
3990 RETURN
3995 REM COMMON ROUTINE
3996 PRINT AT 12,1; FLASH 1;"DOC
UMENT DOES NOT EXIST, RETYPE": P
RINT AT 13,1; FLASH 1;"OR TYPE 9
9 TO EXIT TO SUB MENU": PRINT AT
 14,1; FLASH 1;"AND CHECK CATALO
GE FILE": BEEP 7,-20: PRINT AT 1
2,0;,,,,,,,,
3997 RETURN
4000 REM NEW CHAR SET
4010 BORDER 5: PAPER 7: INK 0: B
RIGHT 1: CLS
4020 PRINT AT 0,5; INVERSE 1;"CH
ARACTER SET"
4030 PRINT AT 10,3;"1  -  NORMAL
";AT 12,3;"2  -  LONG HAND"
4040 INPUT "SELECT OPTION REQUIR
ED ";D
4050 IF D=1 THEN  POKE 23606,0:
POKE 23607,60: GO TO 4800
4100 LET CH=CH+1
4105 IF CH>1 THEN  GO TO 4400
4110 RESTORE 4500
4120 PRINT AT 10,2; FLASH 1;"WAI
T WHILE I POKE NEW CHARS"
4130 FOR F=0 TO 768: POKE 63560+
F,PEEK (15616+F)
4140 IF F/20=INT (F/20) THEN  PR
INT AT 0,0;F
4150 NEXT F
4200 FOR f=520 TO 728: READ a: P
OKE 63560+f,a: NEXT f
4250 FOR f=266 TO 466: READ a: P
OKE 63560+f,a: NEXT f
4400 POKE 23606,72: POKE 23607,2
47
```

```
4500 DATA 0,0,56,4,62,197,60,0,1
6,48,32,124,163,34,60,0,0,0,28,3
2,96,163,28,0,4,4,4,62,69,196,60
,0
4510 DATA 0,0,56,68,249,66,60,0,
12,10,20,112,159,48,80,96,0,0,28
,36,93,134,4,24,0,48,80,96,120,2
03,76,0
4520 DATA 0,16,0,16,49,214,8,0,1
6,0,16,48,211,60,80,96,32,80,116
,164,56,37,38,0,24,40,40,48,80,1
45,14,0
4530 DATA 0,0,104,84,212,85,86,0
,0,0,0,56,100,165,38,0,0,0,56,70
,197,68,56,0,0,0,56,100,167,60,3
2,32
4540 DATA 0,0,56,100,188,5,6,4,0
,0,36,122,161,32,32,0,0,0,48,64,
187,4,56,0,32,32,56,32,97,166,24
,0
4550 DATA 0,0,36,100,164,37,26,0
,0,0,36,38,165,168,16,0,0,0,34,1
06,171,42,20,0,0,0,102,152,136,2
5,102,0
```



```
4560 DATA 0,0,36,100,191,20,36,2
4,0,0,56,72,145,34,116,12
4650 DATA 24,36,36,60,36,165,102
,0,120,36,36,56,37,166,120,0,24,
36,36,32,96,167,24,0
4660 DATA 60,82,81,57,97,209,62,
0,56,72,64,48,64,199,56,0,31,16,
48,95,144,16,16,0
4670 DATA 56,68,68,64,207,68,56,
0,100,36,36,60,36,165,102,0,58,8
4,16,16,16,58,84,0
4680 DATA 63,22,8,8,72,72,48,0,9
8,164,40,48,40,165,98,0,96,160,9
6,32,32,33,126,0
4690 DATA 34,54,42,42,34,163,98,
0,108,178,162,34,34,34,35,0,24,3
6,36,36,101,166,56,0
4700 DATA 60,34,34,124,160,32,32
,0,56,68,68,68,212,76,60,2,56,36
,36,56,100,165,38,0
```

```
4710 DATA 28,34,32,92,130,3,28,0
,113,142,8,8,9,14,24,0,68,68,68,
68,68,197,62,0
4720 DATA 68,68,70,69,168,40,16,
0,130,130,130,147,146,146,108,0,
130,68,40,16,16,170,68,0
4730 DATA 18,18,50,94,130,15,18,
12,62,68,8,16,32,77,242,0
4800 BEEP 2,1
4900 RETURN
5000 REM SAVE DATA
5010 BORDER 5: PAPER 7: INK 0: B
RIGHT 1: CLS
5015 PRINT AT 0,5; INVERSE 1;"SA
VE DATA ROUTINE"
5020 PRINT AT 10,0;"ENTER NAME O
F FILE TO BE SAVED"
5030 INPUT S$
5060 SAVE S$ DATA E$()
5070 PRINT AT 12,0; FLASH 1;"DAT
A SAVE ROUTINE COMPLETED"
5080 BEEP 3,1
5100 RETURN
5500 REM LOAD DATA FILE
```



```
5510 BORDER 5: PAPER 7: INK 0: B
RIGHT 1: CLS
5515 PRINT AT 0,5; INVERSE 1;"LO
AD DATA ROUTINE"
5520 PRINT AT 10,0;"ENTER NAME O
F FILE TO BE LOADED"
5530 INPUT S$
5540 PRINT AT 15,0; FLASH 1;"STA
RT THE TAPE"
5560 LOAD S$ DATA E$()
5570 PRINT AT 15,0; FLASH 1;"STO
P THE TAPE"
5600 BEEP 3,1
5700 RETURN
6000 REM END ROUTINE
6010 BORDER 5: PAPER 7: INK 0: B
RIGHT 1: CLS
6015 PRINT AT 0,5; INVERSE 1;" E
ND ROUTINE "
6020 PRINT AT 10,0;"END OF PROGR
AM"
```

```
6050 INPUT "HAS ALL DATA BEEN SA
VED  Y/N";A$
6060 IF A$(1)="Y" OR A$(1)="y" T
HEN  STOP
6080 IF A$(1)="N" OR A$(1)="n" T
HEN  GO TO 6200
6100 GO TO 6050
6200 RETURN
6500 REM SUB MENU DOC FORMAT
6510 CLS
6520 PRINT AT 0,5;"SUB MENU DOC.
 FORMAT"
6530 PRINT AT 1,5;"==========
=======";
6540 PRINT AT 3,0;"THE FORMAT IS
 ENTERED IN THE    SAME WAY AS Y
OU WOULD ENTER TEXTTHIS INFORMAT
ION IS THEN       OVERPRINTED W
HEN REQUESTED     LIKE A PRE-PR
INTED DOCUMENT    WHICH SAVES Y
OU TIME AND EFFORT."
6545 PRINT AT 9,0;"RE-ENTERING T
HIS ROUTINE BLANKS OUT PREVIOUSL
Y ENTERED FORMAT"
6550 PRINT AT 11,0; INVERSE 1;"P
RESS ENTER TO CONTINUE"
6560 IF INKEY$="" THEN  GO TO 65
60
6570 LET D=31
6590 LET E$(31,641 TO 660)="FORM
AT DOCUMENT"
6600 GO SUB 7782
6610 PAUSE 0
6650 GO SUB 7050
6800 RETURN
7000 REM input text
7010 GO SUB 7700
7013 IF O$<>"Y" AND O$<>"y" THEN
  GO TO 7020
7015 LET T$(1,1 TO 640)=E$(31,1
TO 640)
7020 IF D>30 THEN  GO TO 10
7050 PAPER A(1): INK A(2): BORDE
R A(3): BRIGHT A(4): CLS
7060 PRINT AT 0,0; INVERSE 1;"DO
C ";D;"  ";E$(D,641 TO 660)
7070 PRINT AT 1,0;T$(1,1 TO 640)
7100 LET x=1: LET y=0
7200 IF CODE INKEY$=9 AND x<21 T
HEN  GO SUB 7800: LET y=y+1: BEE
P .01,40
7210 IF CODE INKEY$=8 AND x>0 TH
EN  GO SUB 7800: LET y=y-1: BEEP
 .01,40
7220 IF y>31 AND x<21 THEN  LET
x=x+1: LET y=0
7230 IF y<0 AND x>0 THEN  LET x=
x-1: LET y=31
7240 IF CODE INKEY$=12 THEN  GO
SUB 7600
```

```
7245 IF CODE INKEY$=4 THEN  GO S
UB 7650
7246 IF CODE INKEY$=5 THEN  GO T
O 7560
7250 IF CODE INKEY$=7 AND x<20 T
HEN  GO SUB 7800: LET x=x+1: LET
 y=0
7260 IF CODE INKEY$=10 AND x<20
THEN  GO SUB 7800: LET x=x+1: BE
EP .01,40
7270 IF CODE INKEY$=11 AND x>1 T
HEN  GO SUB 7800: LET x=x-1: BEE
P .01,40
7275 IF CODE INKEY$=6 AND (PEEK
23658=0) THEN  POKE 23658,8: PAU
SE 0: BEEP .2,5
7280 IF CODE INKEY$=6 AND (PEEK
23658=8) THEN  POKE 23658,0: PAU
SE 0: BEEP .2,-12
7290 IF x=21 THEN  LET x=20: LET
 y=31
7300 IF x=0 THEN  LET x=1: LET y
=0
7350 PRINT AT x,y; INK A(2); OVE
R 1;"■"
7380 IF CODE INKEY$=13 THEN  PRI
NT AT 21,0; FLASH 1;"WAIT WHILE
I STORE THIS PAGE": GO SUB 7500:
 BEEP 1,1: GO TO 10
7390 LET I$=INKEY$
7400 IF CODE I$>20 THEN  PRINT A
T x,y;" ": PRINT AT x,y;I$: BEEP
 .05,20: LET T$(1,((x-1)*32+y+1)
)=I$(): LET y=y+1
7450 GO TO 7200
7500 REM store document
7530 LET E$(D,1 TO 640)=T$(1,1 T
O 640)
7550 LET C=0
7560 LET T$(1)=" "
7590 RETURN
7600 REM DELETE CURRENT LINE
7605 LET POS=(X-1)*32+1
7610 LET T$(1,POS TO (640-32))=T
$(1,(POS+32) TO 640)
7610 LET T$(1,POS TO (640-32))=T
$(1,(POS+32) TO 640)
7615 LET T$(1,(640-31) TO 640)="
 "
7620 PRINT AT 1,0;T$(1)
7640 RETURN
7650 REM INSERT LINE
7655 LET POS=(X-1)*32+1
7660 LET T$(1,(POS+32) TO 640)=T
$(1,POS TO (640-32))
7665 LET T$(1,POS TO (POS+31))="
 "
```

```
7670 PRINT AT 1,0;T$(1)
7690 RETURN
7700 REM SUB MENU TEXT INPUT
7710 CLS
7720 PRINT AT 0,5; INVERSE 1;"SU
B MENU TEXT INPUT"
7730 PRINT AT 1,5;"----------
------"
7740 PRINT AT 10,1; FLASH 1;"FOL
LOW INPUT INSTRUCTIONS BELOW"
7750 INPUT "ENTER UNIQUE DOCUMEN
T NO 1-30 ";D
7755 IF D>30 THEN  GO TO 7790
7760 IF E$(D,641 TO 645)<>"
" THEN  PRINT AT 12,1; FLASH 1;"
DOCUMENT ALREADY IN USE RETYPE":
 PRINT AT 13,1; FLASH 1;"OR TYPE
 99 TO EXIT TO MAIN MENU": PRINT
 AT 14,1; FLASH 1;"AND CHECK CAT
ALOGE BY USING THE": PRINT AT 15
,1; FLASH 1;"ENQUIRY SCREEN": BE
EP 7,-20: PRINT AT 12,0;,,,,,,,,
: GO TO 7750
7780 INPUT "ENTER TITLE OF DOCUM
ENT ";E$(D,641 TO 660)
7781 INPUT "DO YOU WANT TO USE T
HE OVERPRINTFEATURE? (Y OR N)";O
$
7782 PRINT AT 12,0;"USE CURSOR K
EYS TO MOVE CURSOR ";AT 13,0;"US
E TRUE VIDEO TO INSERT A LINE";A
T 14,2;"BEFORE THE CURRENT LINE"
;AT 15,0;"USE EDIT TO PLACE CURS
ER AT";AT 16,2;"THE START OF THE
 NEXT LINE"
7783 PRINT AT 17,0;"USE DELETE T
O CLEAR CURRENT LINE";AT 18,0;"U
SE INV. VIDEO TO QUIT"
7784 PRINT AT 19,0;"ONLY TYPE EN
TER ONCE YOU HAVE": PRINT AT 20,
0;"FINISHED YOUR SCREEN": PRINT
AT 21,0; FLASH 1;"PRESS ENTER TO
 CONTINUE": PAUSE 0
7790 RETURN
7800 REM cursor
7805 IF y>31 AND x<21 THEN  LET
x=x+1: LET y=0
7807 IF y<0 AND x>0 THEN  LET x=
x-1: LET y=31
7809 PRINT AT x,y; INK A(2);" "
7810 PRINT AT x,y; INK a(2);T$(1
,(x-1)*32+y+1)
7850 RETURN
8000 STOP
9000 PRINT PEEK 23653+256*PEEK 2
3654
9020 STOP
9700 REM SET DIM SIZE
9710 LET DIMX=31: LET DIMY=660
9750 RETURN
```

INVADED

BY CRAIG SANDERS

Those sneaky aliens are at it again — they've now invaded the 16K Spectrum.

Yes, you've guessed it — we've been got at again by those beasties from outer space! It's really a case of whether you can get the aliens before they can get you. Although there are many space-type games around for the Spectrum, we felt that this particular version was definitely worth including in this magazine. But you shouldn't need too much explanation — just type in the program, don your space-suit and get set to save the universe.

The graphics, variables and introduction were set up at the end of the program to ensure speed for alien movement and player movement.

Notes on Listing

Lines 40-70	Go to the subroutine for graphics, introduction and variables. Define hi-score.
Lines 100-200	Main loop. Ships movement and randomly determine which alien attacks player.
Lines 300-360	Print first alien. GOSUB alien movements. Check if player is killed. Set up points given to that particular alien. Attribute for alien hit routine.
Lines 400-460	Second alien. Return to main loop.
Lines 500-560	Third alien. Return to main loop.
Lines 565-580	Random alien movement for jumping around effect on screen.
Lines 600-650	Lose one life: Print explosion: Set 'alien' to '0' indicating to run lines 170 to 190 and choose new alien.
Lines 1000-1150	Game over. Input name of player if new high score. Return to GOSUB for variables.
Lines 2000-2050	Alien dead. Print explosion: small tune. Add score for that particular alien.
Lines 6000-6200	Set up coordinates depending on the direction that the player's ship is pointing, plot and draw 'fire' and if attribute is not the same as attribute set then GOTO alien killed routine.
Lines 7000-7300	Introduction.
Lines 8000-8050	Set up variables.
Lines 9000-9200	Set up user-defined graphics.

```
 30 REM **16K ZX SPECTRUM**
 40 GO SUB 9000: LET high=0: LE
T a$="": GO SUB 7000
 70 GO SUB 8000
 100 REM **GAME**
 120 PRINT AT 0,0; INK 6;"SCORE=
"; INK 7;score;AT 0,10; INK 6;"H
I-SCORE="; INK 7;high
 125 PRINT AT 21,0; INK 6;"LIVES
-"; INK 7;lives;AT 21,15; INK 6;
"HI-SCORE BY:"; INK 7;a$
 135 PRINT AT c,d; INK 6;CHR$ ch
r
 140 IF INKEY$="7" AND y=1 THEN
 GO SUB 6000
 150 IF INKEY$="8" THEN  LET chr
=chr+1: IF chr=152 THEN  LET chr
=144
 160 IF INKEY$="5" THEN  LET chr
=chr-1: IF chr=143 THEN  LET chr
=151
 165 IF alien  THEN  GO TO 190
 170 LET alien=1: LET rnd=RND
 175 IF rnd<0.3 THEN  LET o=5: L
ET p=1: LET sub=300
 180 IF rnd>=0.3 AND rnd<=0.7 TH
EN  LET o=19: LET p=15: LET sub=
400
 185 IF rnd>0.7 THEN  LET o=5: L
ET p=29: LET sub=500
 190 GO SUB sub
 200 GO TO 135
 300 REM **1st ALIEN**
 310 PRINT AT o,p;" "
 320 GO SUB 565
 340 PRINT AT o,p; INK 5;"0"
 350 IF p=d AND o=c THEN  GO TO
600
 355 LET y=1: LET sc=20: LET att
r=5
 360 RETURN
 400 REM **2nd ALIEN**
 410 PRINT AT o,p;" "
 420 GO SUB 565
 440 PRINT AT o,p; INK 4;"M"
 450 IF p=d AND o=c THEN  GO TO
600
 455 LET y=1: LET sc=50: LET att
r=4
 460 RETURN
 500 REM **3rd ALIEN**
 510 PRINT AT o,p;" "
 520 GO SUB 565
 540 PRINT AT o,p; INK 3;"R"
 550 IF p=d AND o=c THEN  GO TO
600
 555 LET y=1: LET sc=80: LET att
r=3
 560 RETURN
 565 REM **ALIEN MOVEMENT**
 570 LET o=o+(o<c)-(o>c)+INT (RN
D*3-1)
 575 LET p=p+(p<d)-(p>d)+INT (RN
D*3-1)
 580 RETURN
 600 REM **LOSE LIVES**
 610 PRINT AT c,d; INK 7; BRIGHT
 1;"X": FOR f=0 TO 50 STEP 3: BE
EP .006,+f: BEEP .008,+f: NEXT f
: PRINT AT c,d;" "
 620 LET lives=lives-1
 630 IF lives<=0 THEN  GO TO 100
0
 640 PRINT AT o,p;" "
 660 LET alien=0: GO TO 100
1000 REM **GAME OVER**
1010 PRINT AT 21,6; INK 7;"0"
1020 FOR x=0 TO 50: BEEP .05,x:
BEEP .02,x: NEXT x: FOR x=50 TO
0 STEP -3: BEEP .003,x: NEXT x
1030 PRINT AT 10,6; BRIGHT 1; IN
K 4;"G A M E   O V E R"
1040 FOR x=0 TO 100: NEXT x: CLS

1045 PRINT AT 1,3; INK 7; BRIGHT
 1;"S  T  R  A  N  D  E  D "
1050 PRINT AT 5,9;"SCORE=";score
1060 IF score>high THEN  LET hig
h=score
1070 IF score<high THEN  GO TO 1
100
1080 PRINT AT 8,1; INK 7; FLASH
1;"A  N E W   H I G H   S C O R
E": FOR x=0 TO 40 STEP 4: BEEP .
009,x: NEXT x
1085 INPUT  INK 6;"ENTER NAME...
(5 letters)";a$
1090 PRINT AT 11,9; INK 6;"BY ";
a$
1100 PRINT AT 13,7;"HI-SCORE=";h
igh
1110 PRINT AT 19,0; INK 4;"PRESS
 ANY KEY FOR ANOTHER GAME"
1120 PAUSE 0: FOR x=50 TO 0 STEP
 -5: BEEP .005,x: BEEP .005,x: N
EXT x
1130 FOR x=0 TO 21: PRINT AT x,0
;"
  ": BEEP .005,x: NEXT x
1140 INK 7: FOR x=0 TO 50: PLOT
RND*255,RND*170: NEXT x
1150 GO TO 70
2000 REM **KILL ALIEN**
2010 FOR x=50 TO 0 STEP -5: BEEP
 .007,x: NEXT x
```

```
2015 PRINT AT o,p;" ": LET alien
=0
2020 PRINT AT o,p; INK 7;"": FO
R x=0 TO 20: BEEP .009,x: NEXT x
: PRINT AT o,p;" "
2030 LET v=0: LET score=score+sc
2050 GO TO 100
6000 REM **FIRE**
6010 IF chr=151 THEN  LET e=-60:
 LET f=-65
6015 IF chr=150 THEN  LET e=0: L
ET f=-80
6020 IF chr=149 THEN  LET e=60:
LET f=-65
6025 IF chr=148 THEN  LET e=80:
LET f=0
6030 IF chr=147 THEN  LET e=60:
LET f=65
6035 IF chr=146 THEN  LET e=0: L
ET f=70
6040 IF chr=145 THEN  LET e=-62:
 LET f=65
6060 IF chr=144 THEN  LET e=-80:
 LET f=0
6100 INK 6: PLOT g,h: BEEP .006,
3: DRAW e,f
6110 PLOT 115,83: DRAW  OVER 1;e
,f
6120 IF ATTR (o,p)<>attr THEN  G
O TO 2000
6200 RETURN
7000 REM **INTRO'**
7010 BORDER 0: PAPER 0: INK 7: C
LS : FOR f=0 TO 60 STEP 4: BEEP
.008,+f: NEXT f
7015 INK 4: PRINT AT 1,2; BRIGHT
 1;"S   T   R   A   N   D   E
D"
7020 INK 5: PLOT 0,170: DRAW 255
,0: PLOT 0,158: DRAW 255,0: PLOT
 0,154: DRAW 255,0: PLOT 0,170:
DRAW 0,-16: PLOT 255,170: DRAW 0
,-16
7030 INK 6: PRINT ''"This game r
equires the player tofight again
st hordes of alien  craft in yo
ur crippled cruiser  that only h
as rotational thrustsand the abi
lity to shoot weak  energy bolt
s."
7040 PRINT '"Swivel in space des
troying the  waves of endless al
iens before  they have a chance
to dock with your ship and kill
you.".
7050 PRINT '"The player receives
 three lives only,so use them we
ll - attacks come thick and fast
..."
7100 PRINT AT 21,9; INK 7; BRIGH
T 1;"PRESS ANY KEY": PAUSE 0
7120 FOR f=4 TO 21: PRINT AT f,0
;"
  ": BEEP .009,f: NEXT f
7130 PRINT AT 4,0;"There are thr
ee types of alien  craft:";AT 7,
9; INK 5;"  "; INK 6;"- 20 pts";
AT 9,9; INK 4;"  "; INK 6;"- 50
pts";AT 11,9; INK 3;"  "; INK 6;
"- 80 pts"
7140 PRINT AT 14,10; INK 7; BRIG
HT 1;"CONTROLS"
7160 PRINT AT 16,3; INK 6;"'5' -
 left  '8' - right"
7170 INK 6: PRINT AT 19,9;"'0' -
 fire"
7180 INK 7: PRINT #1; BRIGHT 1;"
       PRESS ANY KEY"
7190 PAUSE 0: FOR f=0 TO 50: BEE
P .005,+f: NEXT f
7200 BORDER 0: PAPER 0: INK 7: C
LS
7210 FOR x=0 TO 50: PLOT RND*255
,RND*170: NEXT x
7300 RETURN
8000 REM **VARIABLES**
8010 LET score=0: LET a=146: LET
 chr=a: LET c=11: LET d=14: LET
lives=3: LET g=115: LET h=83: LE
T alien=0: LET v=1
8050 RETURN
9000 REM **GRAPHICS**
9010 FOR f=1 TO 12: FOR g=0 TO 7
: READ a: POKE USR CHR$ (143+f)+
g,a: NEXT g: NEXT f
9020 DATA 2,13,50,196,196,50,13,
3
9030 DATA 192,176,76,67,44,40,16
,16
9040 DATA 24,24,36,36,66,90,165,
195
9050 DATA 3,13,50,194,52,20,8,8
9060 DATA 192,176,76,34,34,76,17
6,192
9070 DATA 8,8,20,52,194,50,13,3
9080 DATA 195,165,90,66,36,36,24
,24
9090 DATA 16,16,40,44,67,76,176,
192
9100 DATA 24,126,219,126,195,129
,195,36
9110 DATA 126,195,231,90,153,90,
129,66
9120 DATA 126,255,231,126,36,231
,129,66
9130 DATA 130,16,0,33,0,136,0,68
9140 FOR x=-10 TO 50 STEP 5: BEE
P .006,x: BEEP .004,x: BEEP .002
,x: BEEP .009,x: NEXT x
9200 RETURN
```

ON FINAL

BY GEORGES BERY

Next time you are in an aircraft, give a thought to the pilot as you come in to land — or give him some advice!

You are in the pilot's seat of a good old transport aircraft, just rolling out from your final turn. You receive by radio, from the tower, the "clear to land", the airfield is dead ahead!

Your job

Start your descent, reduce speed and, if necessary, correcting for wind effect, try to reach the threshold of the runway (distance 0) on the center line, altitude below 10 ft and with a speed around 120 Kms.

Your 16 K ZX81 will then assume you are an expert in touch-down. Keep an eye on your speed during the approach; below 100 Kms, you stall with, as a result, loss of altitude!

An emergency flag, appearing near the gaz indicator, will warn you of any abnormal speed. I wrote gaz thinking to TIMEX, please read throttle!

The controls

As usual, keys 5,6,7 and 8 control respectively the left turn, the climb, the descent and the right turn.

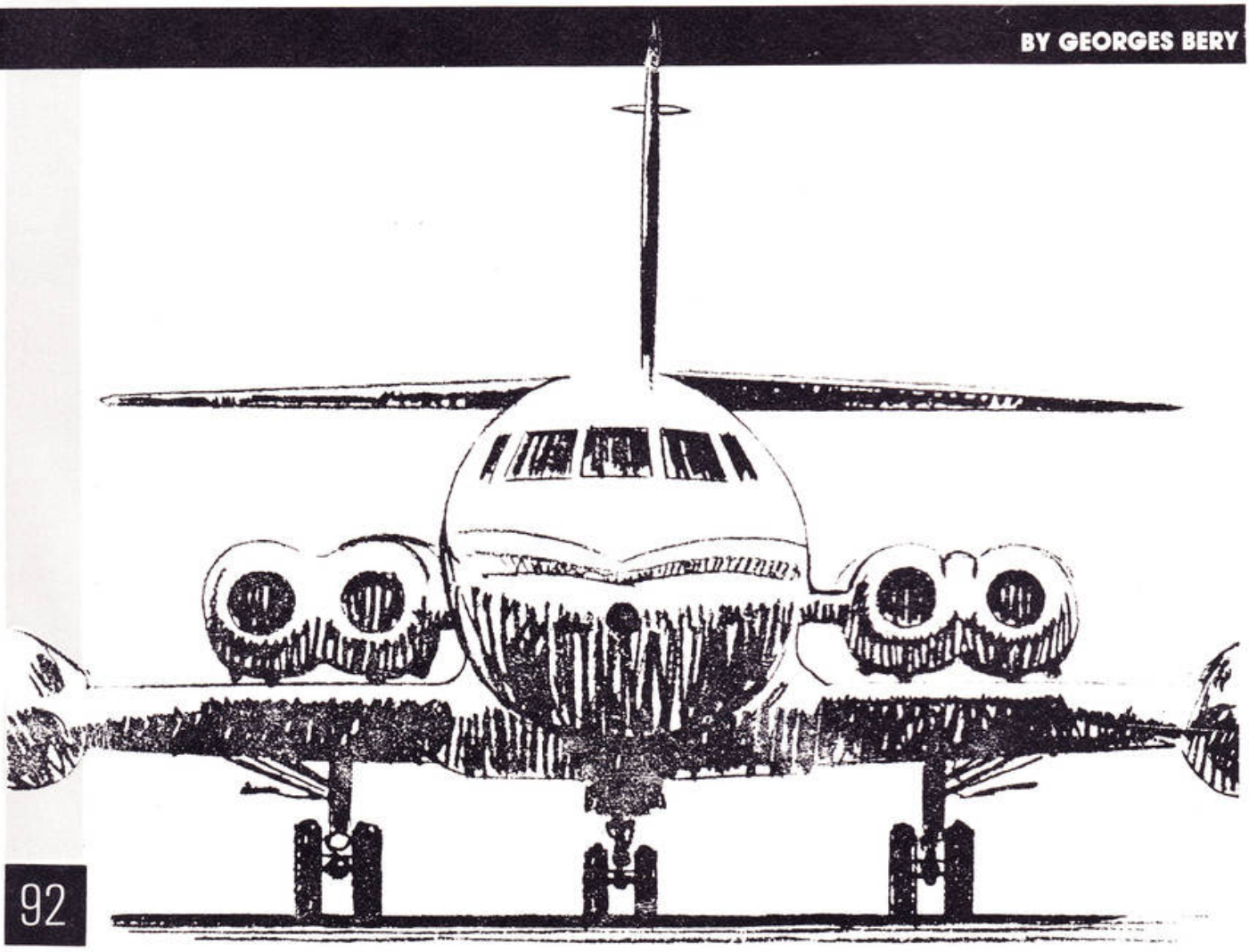
Please, note that keys 6 and 7 don't indicate the action on a stick but the vertical displacement of the aircraft.

Key P increases the power applied to the engine; key M will decrease it. At the beginning of the program, if requested, the pilot's notes will remind you of all the necessary information.

The display

The first picture will give you the choice of consulting or not the pilot's notes. The second picture will ask you to choose the wind strength: 0,1,2 or 3; its direction being left to the choice of the ZX81. After that, you will choose the type of descent: steep or normal.

The combination of the two choices gives 8 levels of difficulty; wind 0 and normal



approach being advised for training.

The cockpit limited panel will then appear:

- at the top of the screen, a grey square indicates the position of the airfield as seen from the cockpit.
- in the center, the artificial horizon indicates the up or down attitude of the aircraft, its direction and rate of turn.
- right or left of the horizon the wind symbol will remind you of the wind direction.
- at the bottom of the screen, you find the instruments: from right to left, the altimeter, the distance meter, the speed indicator and the throttle indicator. Between the two last instruments, an emergency speed flag may appear.

The program

This is nothing else other than a normal ZX81 BASIC program open to any change or addition. For program speed certain functions have been omitted, so, don't hope to find wheels, flaps or fuel . . . I wonder if there is a manner to put them back, in BASIC, in the program without spoiling the speed!

The action on a command key sends the program to an appropriate subroutine which corrects the display and rectifies the variables concerned.

When the altimeter or the distance meter indicates 0, the landing report appears on the screen. A crash will send you to subroutine 1850 for a good shake!

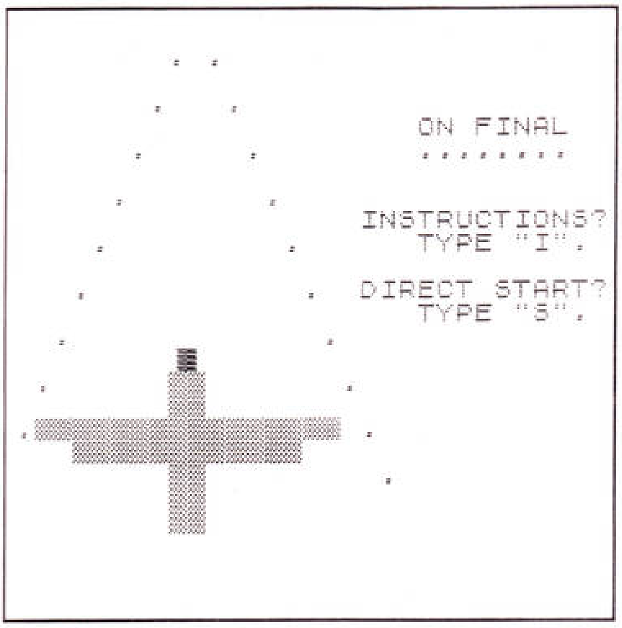
Last remark

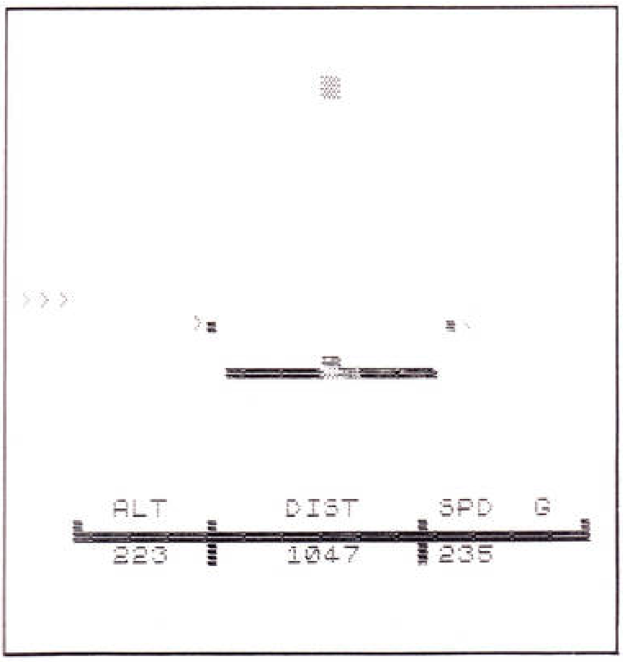
The computer will only execute one order at a time (no use pushing two keys together). In case of emergency, this feature requires sound judgement and quick reactions to avoid a crash.

Now, to all ZX81 PILOTS, good training and lots of happy landings.

The screen dump

The airfield is dead ahead, the wind blows from left, the aircraft is in descent, turning to the left. The other indications are self-explanatory.






```
   1 LET X=1
   5 LET B=10
  20 FOR A=0 TO 18 STEP 2
  30 PRINT AT A,B+X;".";AT A,B-X;"."
  40 LET X=X+1
  50 NEXT A
  60 FOR A=14 TO 20
  70 PRINT AT A,B-1;"▒▒"
  80 NEXT A
  90 PRINT AT 13,B-1;"█"
 100 PRINT AT 16,2;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 110 PRINT AT 17,4;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 120 PRINT AT 3,22;"ON FINAL"
 130 PRINT AT 4,22;"........"
 140 PRINT AT 7,19;"INSTRUCTIONS?"
 150 PRINT AT 8,22;"TYPE ""I""."
 160 PRINT AT 10,19;"DIRECT START?"
 170 PRINT AT 11,22;"TYPE ""S""."
 180 IF INKEY$="I" THEN GOTO 1000
 190 IF INKEY$="S" THEN CLS
 200 IF INKEY$="S" THEN GOTO 1120
 210 IF INKEY$<>"S" OR INKEY$<>"I" THEN GOTO 180
 300 CLS
 305 PRINT AT 19,5;"ALT";AT 19,14;"DIST";AT 19,22;"SPD  G"
 310 PRINT AT 20,3;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 320 PRINT AT 21,10;"▌ ▌"
 325 PRINT AT 11,9;">▖▄▄▄▄▄▄▄▄▄▄▖<"
 330 LET A=11
 335 LET B=A
 340 LET C=A
 345 LET D=1500
 350 LET F=16
 355 LET U=0
 360 LET V=200
 365 LET J=1+INT (RND*2)
 370 IF J=2 THEN LET W=-W
 375 IF J=2 THEN PRINT AT 10,0;">>>"
```

```
 380 IF J=1 THEN PRINT AT 10,28;
"<<<"
 400 IF INKEY$="" THEN GOTO 440
 405 IF INKEY$="6" THEN GOSUB 15
00
 410 IF INKEY$="7" THEN GOSUB 15
50
 415 IF INKEY$="5" THEN GOSUB 16
00
 420 IF INKEY$="8" THEN GOSUB 16
50
 425 IF INKEY$="P" THEN GOSUB 17
00
 430 IF INKEY$="M" THEN GOSUB 17
50
 440 LET F=F-B+11+W/10
 445 IF F<=1 THEN LET F=1
 450 IF F>=30 THEN LET F=30
 455 PRINT AT 1,F-5;"      ▒
"
 460 LET V=V-(11-A)+U
 465 PRINT AT 21,22;V;" "
 470 IF V<100 OR V>250 THEN GOSU
B 1800
 475 LET D=D-(V/10)
 480 PRINT AT 21,14;INT D;" "
 485 IF D<=0 THEN GOTO 550
 490 LET H=H+((11-A)*V/100)
 495 PRINT AT 21,5;INT H;" "
 500 IF H<=0 THEN GOTO 600
 505 GOTO 400
 550 IF H<10 AND F>=14 AND F<=18
 THEN PRINT AT 3,8;" GOOD LANDIN
G "
 555 IF H>=10 THEN PRINT AT 3,5;
" OVER BASE TOO HIGHT "
 600 IF D>0 THEN PRINT AT 3,12;"
 CRASH "
 605 IF D>0 THEN GOSUB 1850
 610 IF D>=-15 AND F>=14 AND F<=
18 AND D<1 AND H<10 THEN PRINT A
T 3,8;" GOOD LANDING "
 615 IF F<14 OR F>18 THEN PRINT
AT 4,8;" OUT OF RUNWAY "
 620 IF V>140 THEN PRINT AT 5,10
;" TOO FAST "
 625 IF V>200 THEN PRINT AT 6,4;
" MUCH TOO FAST.. CRASH "
 630 IF V>200 THEN GOSUB 1850
 635 PAUSE 250
 640 FOR R=0 TO 18
 641 PRINT AT R,0;"
                  *  "
 642 NEXT R
 650 PRINT AT 9,0;"ANOTHER LANDI
NG? (Y/N)"
 655 IF INKEY$="" THEN GOTO 650
 660 IF INKEY$="Y" THEN CLS
 665 IF INKEY$="Y" THEN GOTO 112
0
 670 IF INKEY$="N" THEN CLS
 675 IF INKEY$="N" THEN PRINT AT
 10,0;"DON""T FORGET THE AIRPORT
""S TAXES"
 680 STOP
1000 CLS
1005 FOR A=0 TO 20 STEP 2
1010 PRINT AT A,0;"O";AT A,31;"O
"
1020 NEXT A
1030 PRINT AT 0,2;"PILOT""S NOTE
S."
1040 PRINT AT 2,2;"CLIMB: KEY 7"
1050 PRINT AT 3,2;"DESCEND: KEY
 6"
1060 PRINT AT 4,2;"RIGHT TURN: K
EY 8"
1070 PRINT AT 5,2;"LEFT TURN: KE
Y 5"
```

```
1080 PRINT AT 7,2;"TO REDUCE GAS
: KEY M"
1090 PRINT AT 8,2;"TO INCREASE G
AS: KEY P"
1100 PRINT AT 10,2;"SPEED LIMITS
: 100  TO  250"
1110 PRINT AT 11,2;"LANDING SPEE
D: + - 120"
1115 PRINT AT 12,2;"GAS LIMITS:
-5  TO  +5"
1120 PRINT AT 14,2;"WIND EFFECT:
"
1121 PRINT AT 15,12;" INPUT 0,1,
2 OR 3"
1130 PRINT AT 16,2;"WIND EFFECT=
"
1140 INPUT W
1145 IF W<0 OR W>3 THEN GOTO 112
0
1150 PRINT AT 18,2;"TYPE OF APPR
OACH:"
1151 PRINT AT 19,12;"1=STEEP 2=N
ORMAL"
1160 PRINT AT 20,2;"YOUR CHOICE=
"
1170 INPUT G
1180 IF G=1 THEN LET H=500
1190 IF G=2 THEN LET H=300
1200 IF G<1 OR G>2 THEN GOTO 116
0
1220 GOTO 300
1500 PRINT AT A,11;"              "
1505 LET A=A+1
1510 IF A>=16 THEN LET A=16
1515 PRINT AT A,11;"▄▄▄▄▄▄▄▄▄▄▄▄▄"
1520 RETURN
1550 PRINT AT A,11;"              "
1555 LET A=A-1
1560 IF A<=6 THEN LET A=6
1565 PRINT AT A,11;"▄▄▄▄▄▄▄▄▄▄▄▄▄"
1570 RETURN
1600 PRINT AT C,22;" ";AT B,10;"
 "
1605 LET B=B-1
1610 LET C=C+1
1615 IF B<=6 THEN LET B=6
1620 IF C>=16 THEN LET C=16
1625 PRINT AT B,10;"▪";AT C,22;"
▪"
1630 RETURN
1650 PRINT AT C,22;" ";AT B,10;"
 "
1655 LET C=C-1
1660 LET B=B+1
1665 IF C<=6 THEN LET C=6
1670 IF B>=16 THEN LET B=16
1675 PRINT AT B,10;"▪";AT C,22;"
▪"
1680 RETURN
1700 LET U=U+1
1705 IF U>=5 THEN LET U=5
1710 PRINT AT 21,27;U;" "
1715 RETURN
1750 LET U=U-1
1755 IF U<=-5 THEN LET U=-5
1760 PRINT AT 21,27;U;" "
1765 RETURN
1800 IF V<=100 THEN PRINT AT 21,
25;"S"
1805 IF V>=250 THEN PRINT AT 21,
25;"F"
1810 IF V<95 THEN LET H=H-10
1815 IF V<=100 AND V>94 THEN LET
 H=H-5
1825 RETURN
1850 FOR N=0 TO 50
1855 FAST
1860 PAUSE 2
1865 SLOW
1870 NEXT N
1875 RETURN
2000 SAVE "ON FINA▒"
2010 RUN
```

PHOTOCOPIES

ZX COMPUTING

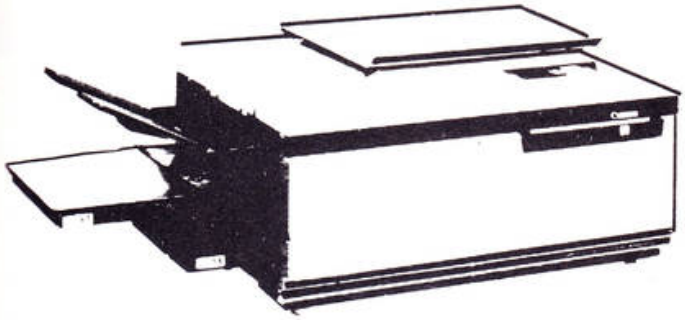
To take advantage of this simple service, just fill in the required information and send it (or a photocopy) to:

ZX Computing Photocopies,
No. 1, Golden Square,
London W1R 3AB.

together with your money and we'll do the rest!

Lost and can't be replaced? Well, if you've lost one of the early issues that cannot be replaced from our stocks of backnumbers, all is not quite lost.

If you know the article name and the issue it appeared in, we can supply you with a photocopy for the miserly sum of £1.50 including postage and packing.



ZX PHOTOCOPIES

NAME .....

ADDRESS .....

.....

POSTCODE .....

Please send me Photocopies of the following items

ISSUE	ARTICLE	PAGES

At £1.50 each, I enclose £ .....

Cheques and Postal Orders should be made payable to ASP Ltd.

EXTRA GRAPHICS

Now anyone can illustrate their programs with this listing for extra graphic characters for the Spectrum

BY I. J. ROGERS

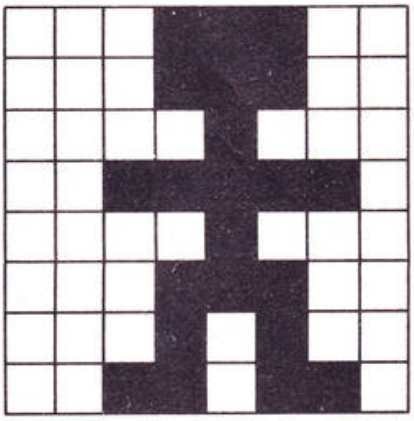
Let us say we want to store the new graphics in memory from 40000 and onwards. First we shall look at how we shall store the new characters in memory. Each character can be put into an 8×8 grid. Each row in the grid can be written as an 8 digit binary number. When each binary number is converted into decimal it will be between 0 and 255.

We will take the binary number 01101011 and convert it into decimal:

Value in decimal	128 64 32 16 8 4 2 1
Binary number	0 1 1 0 1 0 1 1

If there is a 1 in the Binary number, then in decimal it will be the number above. So: The first 1 is under 64, the 2nd under 32, the 3rd under 8, the 4th under 2 and the 5th under 1.
64+32+8+2+1=107.

Below is a graphic man, with the binary and decimal numbers given.



Row 1	BIN 00011100	28
Row 2	BIN 00011100	28
Row 3	BIN 00001000	8
Row 4	BIN 00111110	62
Row 5	BIN 00001000	8
Row 6	BIN 00011100	28
Row 7	BIN 00010100	20
Row 8	BIN 00110110	54

We will POKE each number for the 1st extra graphic into memory locations 40000 to 40007 inclusive. (The second extra graphic would be stored in memory locations 40008 to 40015, the third from 40016 to 40023 and so on).

Type in this program to POKE the extra graphics into memory from 40000 and onwards. (= space).

LISTING 1

```
9000    LET G=1
9005    FOR F=1 TO 96
9010    FOR L=39999+G TO 39999+G+7
9020    INPUT "GRAPHIC ";(F); " ";H
9025    IF H<0 OR H>255 THEN GOTO 9090
9030    POKE L,H
9040    NEXT L
9050    LET G=G+8
9060    NEXT F
9070    PRINT "YOU HAVE USED UP ALL OF THE 96 EXTRA GRAPHICS"
9080    STOP
9090    PRINT "NUMBER IS NOT IN THE RANGE OF 0-255. ENTER AGAIN":
        PAUSE 0: CLS: GOTO 9020
```

To STOP the program, type in STOP when you are asked to INPUT the binary numbers for graphic. To SAVE the extra graphics, use:

SAVE "GRAPHICS" CODE 40000, number of extra graphics *8

To LOAD, type:

LOAD " " CODE

Type in Listing 2:

```
LISTING 2
9000  POKE 23606, 39744-256* INT (39744/256)
9010  POKE 23607, INT (39744/256)
9998  STOP
9999  POKE 23606,0: POKE 23607,60
```

In lines 9000 and 9010, the 39744 is obtained by taking away 256 from 40000. With the extra graphics in memory at location 40000 RUN Listing 2. The extra graphics are used in place of the character set. See Table 1

As this only effects part of the character set, graphics A—U and the predefined graphics can still be used. To get back to the normal character set use GOTO 9999. To SAVE User-defined graphics, use:

SAVE "GRAPHICS" CODE 65368, 168

To LOAD, use:

LOAD " " CODE.

```
LISTING 3
10:  LOAD " " CODE: REM LOAD EXTRA GRAPHICS
20:  LOAD " " CODE: REM LOAD UDG's
30:  LOAD " ": REM LOAD MAIN PROGRAM
```

I would suggest that you SAVE listing 3, (or something similar) by SAVE "GRAPHICS" LINE 0. Then SAVE the extra graphics with SAVE "GRAPHICS" CODE 40000, no. of graphics *8. SAVE the normal UDG's with SAVE "GRAPHICS" CODE 65368,168. Then SAVE the main listing with SAVE "PROGRAM" LINE O.

Remember, to use the extra graphics in the main listing, you should include lines 9000 and 9010 of listing 2, and to get back to the normal character set, you should include somewhere line 9999 of listing 2.

We can have another 96 extra graphics, but it would simplify things if they were not used with the first set of extra graphics. Getting the second set of extra graphics is done in a similar way to the first. All we are really doing is changing the memory location at which we store them.

Storing the first set of extra graphics took up 768 addresses in the computer's memory. (8×96=768). We can store the second set of extra graphics in memory from 40768 (40000+768). We will POKE the new codes for extra graphics into memory from address 40768 and onwards by altering some of the lines in listing 1.

Change the following lines:

9010 FOR L=40767+G TO 40767+G+

Table 1 — In order — the characters that the extra graphics replace.

Code	Character	Code	Character
32	space	66	B
33	!	67	C
34	"	68	D
35	#	69	E
36	$	70	F
37	%	71	G
38	&	72	H
39	'	73	I
40	(	74	J
41	)	75	K
42	*	76	L
43	+	77	M
44	,	78	N
45	—	79	O
46	.	80	P
47	/	81	Q
48	0	82	R
49	1	83	S
50	2	84	T
51	3	85	U
52	4	86	V
53	5	87	W
54	6	88	X
55	7	89	Y
56	8	90	Z
57	9	91	[
58	:	92	/
59	;	93	]
60	<	94	↑
61	=	95	—
62	>	96	£
63	?	97	a
64	@	98	b
65	A	99	c
100	d	114	r
101	e	115	s
102	f	116	t
103	g	117	u
104	h	118	v
105	i	119	w
106	j	120	x
107	k	121	y
108	l	122	z
109	m	123	{
110	n	124	j
111	o	125	}
112	p	126	~
113	q	127	©

Enter the codes in the same way as you did before. To SAVE the second set of extra graphics, use:

SAVE "GRAPHICS" CODE 40768, number of extra graphics *8.

To LOAD, type:

LOAD " " CODE

To use the 2nd set of extra graphics, change the following lines in listing 2:

9000 POKE 23606, 40512-256 * INT (40512/256)
9010 POKE 23607, INT (40512/256)

Also add an appropriate line to listing 3 to LOAD the second set of extra graphics.

You can add as many sets of extra graphics as you wish until there is no more room in memory, but don't forget to alter listings 1, 2 and 3 where appropriate.

Explanation

On page 173 of your manual, you will see that addresses 23606 and 23607 (CHARS) is 256 less than the address of the character set. To find out the value, we use PEEK 23606+256* PEEK 23607. This is 15360 we add 256 and get 15616. 15616 is the normal address of the character set from space to©. Each character takes up 8 addresses in the computer's memory. So if we PEEK the address 15616 to 15623, we will get the decimal numbers for each row of the 2nd character (!)

What we want to do though, is to POKE the new character set into a different part of memory, in this case, address 40000 was used. When we have done that, we want to change the address stored at 23606 and 23607 to the address of the new character set (40000). Therefore, we take 256 away from 40000 and we want PEEK 23606+256 * PEEK 23607 to be 39744. (39744 is 40000-256).

So we used:
POKE 23606, 39744-256 * INT (39744/256)
POKE 23607, INT (39744/256)

as that is the method suggested half-way down page 173 of the manual.

When we want to use the new character set, we change the address of where the character set is stored in memory. You can store as many character sets as you wish in memory, but I would suggest that you only use one at a time.

Finally, I include a list of useful POKES.

POKE 23692,255 — Scrolls the screen up one line
POKE 23658,255 — Upper Case lock
POKE 23658,0 — Lower Case lock
POKE 23756,0 — Changes first line of program to 0
POKE 23625, line no.–256* INT (line no./256) — POKEing both numbers will move program cursor to specified line.
POKE 23626, INT (line no./256)

POKE 23618, line no.–256* INT (line no./236) — By POKEing all these 3 addresses you do not have to start the program at the first statement in the line.
POKE 23619, INT (line no./256)
POKE 23620, number of statement in line.

ie If line 20 was:
20 PRINT "Instructions...": PAUSE 0: CLS
then POKE 23618 and 23619 to store line no. POKE 23620,2
then program will start at the PAUSE 0 in line 20.

POKE 23659,0 —
Makes program unbreakable but will not allow bottom lines of screen to be used for INPUT etc.

ROM ROUTINES

BY NORMAN STEVENS

Some useful subroutines to polish up your programming

These subroutines, which will probably be of more use to the experienced machine code programmer, will help you to add the professional finishing touches to your programs.

Printing subroutines

Scores etc. can be conveniently printed using the routine at 2725. Calling this routine will result in the two byte number occupying bytes (hl) and (hl+1) being printed in the four spaces following the current print position (as defined by S.POSN and DF.CC) with loading spaces if the number is less than 1000. The number to be printed must be stored in the non-standard order of (high byte), (low byte). The current print position can be set by the subroutine at 2293. This sets the column number to the value of the C register and the line number to that of the b register, calculates, and then sets the print position to DF.CC.

Thus the following will print the line number of the first BASIC program line at the print position 0,28 (top right of the screen):

```
ld bc,NN   28,0    ;  1,28,0
call NN    2293    ;  205,24,8
ld hl,NN   16509   ;  33,125,4
call NN    2725    ;  205,165,10
ret                   201
```

Printing a number already in the registers

This is more difficult because of the way in which the subroutine 2725 uses the stack. The following subroutine will print the number in the de registers at the position given by the bc registers, preserving the de register values and returning after the subroutine call in the normal way.

```
push de      213
call 2293    205,24,8
pop de       209
push de      213
jp 2729      195,169,0
```

Character strings (including keywords and tokens) can be printed using the subroutine at 2923. On calling this subroutine, bc must be the length of the string to be printed (and this may be zero), and de the address of the first character in the string. The string is printed at the current print position. Single characters (excluding keywords and tokens) can be printed at the current print position using the RST 16 command. The code of the character is taken from a register.

All three subroutines increase the current print position to the column following the last character printed. If the USR routine is called from a LLIST or LPRINT statement, then the output will be LPRINTed rather than PRINTed.

INKEY$

This subroutine will return with the register containing CODE INKEY$.

```
call 699      205,187,2
ld b,h        68
ld c,l        77
ld d,c        81
inc d         20
call nz 1981  196,189,7
ld a,d        122
ret n,c       208
ld a(hl)      126
ret           201
```

Returning from the USR routine

It sometimes happens that, depending on the results of the calculations performed, one wishes to continue processing at different places in the BASIC program. One could use the value returned of the bc register pair as a flag, but this may make the passing of information to the BASIC system more difficult. A neat way of accomplishing this is to use the subroutine at 3718. This sets the variables NXTLIN to the address of the line whose number is stored in the hl registers. Thus, after executing the line containing the USR function, the next line processed will effectively be GOTO HL, where HL is the value of the hl registers at the time (3718) was last called.

MISSILE CONTROL

Your city is under attack — can you hit the missiles first?

War has broken out and your city is under attack from the enemy. His missiles home-in on the city, but they're not easy to hit. This great implementation of the arcade game will test your skills to the fullest so get ready for Battle Stations.

```
  1 REM ************************
        *Underlined characters*
        *are entered in       *
        *GRAPHICS mode.       *
        ************************
  5 BORDER 0: PAPER 0: INK 7
  9 REM U.D.G.s
 10  FOR n=144 TO 147
 20 FOR f=0 TO 7
 30 READ a
 40 POKE USR CHR$ n+f,a
 50 NEXT f
 60 NEXT n
 70 DATA 64,64,96,98,226,243,25
5,255
 80 DATA 0,10,10,26,26,31,255,2
55
 90 DATA 1,3,7,15,31,63,127,255
100 DATA 128,192,224,240,248,25
2,254,255
105 REM SET UP VARIABLES
110 LET h=0
200 LET a=10: LET b=15: LET a1=
a: LET b1=b: LET nm=2: LET sc=0
```

```
210 DIM c(4): DIM d(20): DIM e(
20): DIM m(3): DIM t(3)
220 FOR f=1 TO 4: LET c(f)=0: N
EXT f
225 CLS : PRINT AT 10,7;"MISSIL
E CONTROL";AT 12,5;" by Michael
Harley "
230 INPUT "Do you want instruct
ions ?";a$
240 IF a$="y" OR a$="Y" THEN  G
O SUB 2000
300 CLS
301 LET k=0: FOR f=1 TO 4: IF c
(f)=1 THEN  LET k=k+1
302 NEXT f: IF k=4 THEN  GO TO
1000: REM CHECK IF ANY CITIES AR
E HIT
310 PRINT AT 20,0; INK 6;"
"
320 FOR f=-45 TO 45 STEP 5: BEE
P .2,f: NEXT f
330 FOR f=1 TO nm: LET e(f)=175
: LET d(f)=INT (RND*190)+26: NEX
T f
340 LET m(1)=5: LET m(2)=5: LET
```

BY MICHAEL HARLEY

```
 m(3)=5: LET t(1)=31: LET t(2)=1
27: LET t(3)=223
 345 REM  PRINT CITIES AND BASES
 350 PRINT AT 19,3; INK 3;"CD";A
T 19,15;"CD";AT 19,27; INK 3;"CD
"
 360 IF c(1)=0 THEN  PRINT AT 19
,7; INK 1;"BB"
 370 IF c(2)=0 THEN  PRINT AT 19
,10; INK 1;"BB"
 380 IF c(3)=0 THEN  PRINT AT 19
,20; INK 1;"BB"
 390 IF c(4)=0 THEN  PRINT AT 19
,23; INK 1;"BB"
 395 REM CHECK POSITION OF SIGHT
 400 IF b<1 THEN  LET b=1
 401 IF b>30 THEN  LET b=30
 402 IF a<2 THEN  LET a=2
 403 IF a>17 THEN  LET a=17
 410 REM ARE MISSILES FINISHED T
HEIR ATTACK ?
 420 LET chk=0
 430 FOR f=1 TO nm: IF e(f)=0 TH
EN  GO TO 472
 435 IF e(f)<16 THEN  GO TO 900
 440 IF d(f)<15 THEN  LET d(f)=1
5
 450 IF d(f)>240 THEN  LET d(f)=
240
 455 IF RND>.6 THEN  GO SUB 800
 460 IF e(f)<=23 THEN  GO SUB 70
0
 466 INK 2: PLOT d(f),e(f): PLOT
 d(f)+1,e(f): LET e(f)=e(f)-1: L
ET d(f)=d(f)+INT (RND*2)-INT (RN
D*2): INK 0
 470 GO TO 480
 472 FOR l=1 TO nm: LET chk=chk+
e(l)
 474 IF chk<>0 THEN  GO TO 480
 476 NEXT l
 478 GO TO 900
 480 NEXT f
 482 LET b$=INKEY$
 485 REM KEYBOARD INPUT
 490 IF b$="1" OR b$="2" OR b$="
3" THEN  GO TO 650
 500 IF b$="u" OR b$="U" THEN  L
ET a=a-1
 510 IF b$="i" OR b$="I" THEN  L
ET a=a+1
 520 IF b$="o" OR b$="O" THEN  L
ET b=b-1
 530 IF b$="p" OR b$="P" THEN  L
ET b=b+1
 535 REM PRINT SIGHT
 540 PRINT AT a1,b1; INK 2;" "
 550 PRINT AT a,b; INK 6;"+": IN
K 0
 555 LET a1=a: LET b1=b
 560 BEEP .05,-30
 570 GO TO 400
 600 REM  FIRE ROUTINE
 650 LET x=VAL b$: IF m(x)=0 THE
N  GO TO 500
 652 LET b2=b*8-t(x)+4: LET a2=(
21-a)*8-20
 654 BEEP .3,-40: INK 2: PLOT t(
x),24: DRAW b2,a2: PLOT  OVER 1;
t(x),24: DRAW  OVER 1;b2,a2
 656 PRINT AT a,b; INK 6;"+"
 658 LET m(x)=m(x)-1
 660 FOR n=1 TO 4: INK 7: IF n-3
<=0 THEN  INK 2
 662 CIRCLE b2+t(x),a2+24,n: NEX
T n
 664 FOR n=1 TO 4: CIRCLE  OVER
1;b2+t(x),a2+24,n: NEXT n
 666 PRINT AT a+1,b;" ";AT a-1,b
;" "
 668 FOR z=1 TO nm
 670 IF d(z)>b*8+2 AND d(z)<b*8+
6 AND e(z)>(21-a)+2*8 AND e(z)<(
22-a)*8-2 THEN  LET e(z)=0: LET
sc=sc+10
 672 NEXT z
```

```
 680 GO TO 400
 699 REM CHECK IF MISSILES HIT T
ARGET
 700 IF d(f)>23 AND d(f)<40 THEN
  LET m(1)=0: PRINT AT 19,3;"  "
: LET e(f)=0: RETURN
 710 IF d(f)>55 AND d(f)<72 THEN
  LET c(1)=1: PRINT AT 19,7;"  "
: LET e(f)=0: RETURN
 720 IF d(f)>79 AND d(f)<96 THEN
 LET c(2)=1: PRINT AT 19,10;"
": LET e(f)=0: RETURN
 730 IF d(f)>119 AND d(f)<136 TH
EN  LET m(2)=0: PRINT AT 19,15;"
  ": LET e(f)=0: RETURN
 740 IF d(f)>159 AND d(f)<176 TH
EN  LET c(3)=1: PRINT AT 19,20;"
  ": LET e(f)=0: RETURN
 750 IF d(f)>183 AND d(f)<200 TH
EN  LET c(4)=1: PRINT AT 19,23;"
  ": LET e(f)=0: RETURN
 760 IF d(f)>215 AND d(f)<232 TH
EN  LET m(3)=0: PRINT AT 19,27;"
  ": LET e(f)=0: RETURN
 780 RETURN
 799 REM MISSILE 'HOME IN' ROUTI
NE
 800 IF d(f)>55 AND d(f)<72 THEN
  RETURN
 810 IF d(f)>79 AND d(f)<96 THEN
  RETURN
 820 IF d(f)>159 AND d(f)<176 TH
EN  RETURN
 830 IF d(f)>183 AND d(f)<200 TH
EN  RETURN
 840 IF d(f)<56 THEN  LET d(f)=d
(f)+3
 850 IF d(f)>95 AND d(f)<127 THE
N  LET d(f)=d(f)-3
 860 IF d(f)>126 AND d(f)<160 TH
EN  LET d(f)=d(f)+3
 870 IF d(f)>200 THEN  LET d(f)=
d(f)-3
 880 RETURN
 899 REM ROUND END AND BONUS SCO
RES
 900 IF nm<20 THEN  LET nm=nm+1
 910 LET sc=sc+5*m(1)+5*m(2)+5*m
(3)
 920 LET sc=sc+50*(c(1)=0)+50*(c
(2)=0)+50*(c(3)=0)+50*(c(4)=0)
 930 GO TO 300
 999 REM END ROUTINE
1000 PAPER 2: INK 7: CLS : FLASH
 1
1100 PRINT AT 6,8;"
   "
    1110 PRINT AT 7,8;" ▀▀▀▀▀ ▀
    ▀ ▀▀ ▀▀ "
    1120 PRINT AT 8,8;"   ▀   ▀
    ▀ ▀   "
```

```
1130 PRINT AT 9,8;"   ▀   ▀▀▀ ▀▀
▀  "
1140 PRINT AT 10,8;"   ▀   ▀ ▀ ▀
   "
1150 PRINT AT 11,8;"   ▀   ▀ ▀ ▀
▀▀▀ "
1155 PRINT AT 12,8;"
   "
1160 PRINT AT 13,8;" ▀▀▀▀ ▀   ▀ ▀
▀▀  "
1170 PRINT AT 14,8;" ▀     ▀▀ ▀ ▀
  ▀ "
1180 PRINT AT 15,8;" ▀▀▀  ▀ ▀▀ ▀
  ▀ "
1190 PRINT AT 16,8;" ▀     ▀  ▀ ▀
  ▀ "
1200 PRINT AT 17,8;" ▀▀▀▀ ▀   ▀ ▀
▀▀  "
1202 PRINT AT 18,8;"
   "
1205 RESTORE 1220
1210 FOR s=1 TO 11: READ a,b: BE
EP a,b: NEXT s
1220 DATA .75,-8,.75,-8,.4,-8,.7
5,-8,.75,-5,.5,-6,.5,-6,.5,-8,.5
,-8,.75,-9,.75,-8
1230 FLASH 0: CLS
1240 PRINT AT 10,10; INVERSE 1;"
Score"; INVERSE 0;" ";sc
1250 IF h<sc THEN  LET h=sc
1260 PRINT AT 12,15; INVERSE 1;"
Hi-score"; INVERSE 0;" ";h
1270 INPUT "Do you want another
game ?";a$
1280 IF a$="y" OR a$="Y" THEN  G
O TO 1300
1290 STOP
1300 CLS
1310 GO TO 200
1999 REM INSTRUCTIONS
2000 PAPER 5: INK 9: CLS
2010 PRINT AT 2,3;" '1','2','3':
-FIRE"
2020 PRINT AT 4,8;"U:- Up       O
:-Right"
2030 PRINT AT 5,8;"I:- Down
P:-Left"
2040 PRINT AT 8,0;"   Stop the a
lien missiles from destroying yo
ur cities.You       have three ba
ses which have fivemissiles that
 you can use."
2050 PRINT "   Use the keys as g
iven above"
2060 PRINT "   Move the cross on
to your      target,it will flash
 red if you are very close but,t
his will notalways mean a hit"
2070 PRINT AT 20,0;"Press any ke
y to start": PAUSE 4e4
2080 RETURN
```

MATHS MAZE

BY J.S. THURLBY

Learning can be fun. So they used to tell us! This program allows you to play an adventure game with a difference. Brains at the ready, folks!

In this educational graphic adventure for the 48K Spectrum the player has to guide his man around the maze to find six pieces of treasure. After a room has been entered the player must remain there until he has correctly solved a mathematical problem. There are three ways of playing the game, involving either addition, subtraction or multiplication problems, and there are 9 levels of skill. These variations make it suitable for a wide range of player's ability.

Playing the game

On the left of the screen is a bird's eye view of the player's position in the maze. The right hand side of the screen carries a constant progress display, indicating lives lost and treasures found. The lower three lines of the screen are used for instructions/questions and input to the program.

The program has an auto-run/save routine that can be initiated by typing RUN 9900. During the course of the game the program can be aborted by pressing SYMBOL SHIFT and A together. This exits from the maze and gives the player the option to save the current position to tape or microsave. At the start of the game there is an option which allows a previous game to be reloaded. Loading and saving of an old game only takes a few seconds due to the use of a method which involves only storing the character array containing the important details, such as current score, lives left, position in the maze etc, rather than saving the whole program.

Subroutines

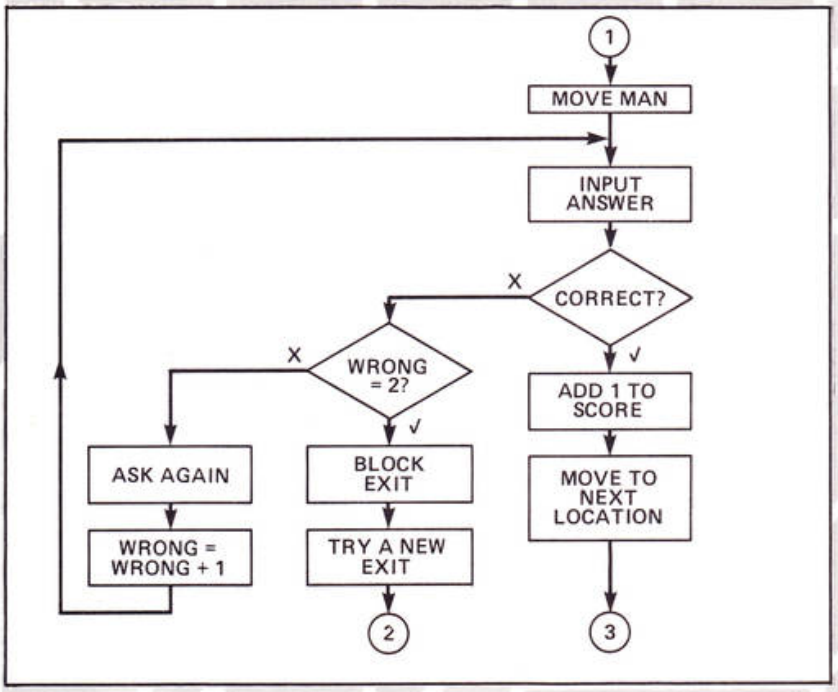
The program is written in subroutines called from a 'main' program at the start.
The lines are:

4000-4100	Select level
4200-4400	Facility to restart a previous game
4500-4990	Initial display
5500-5590	Lose a life
6000-6590	Move a man
7000-7130	Draw a location
7140-7280	Calculate sums
7500-7710	Abort game
8000-8250	Initialise UDG's, Maze
8270-8370	Set up screen display
8500-8540	Move man to pick up treasure
9000-9290	Data for maze definition
9500-9690	End of game display
9700-9830	Game over
9900-9960	Save game and auto-run

Program variables

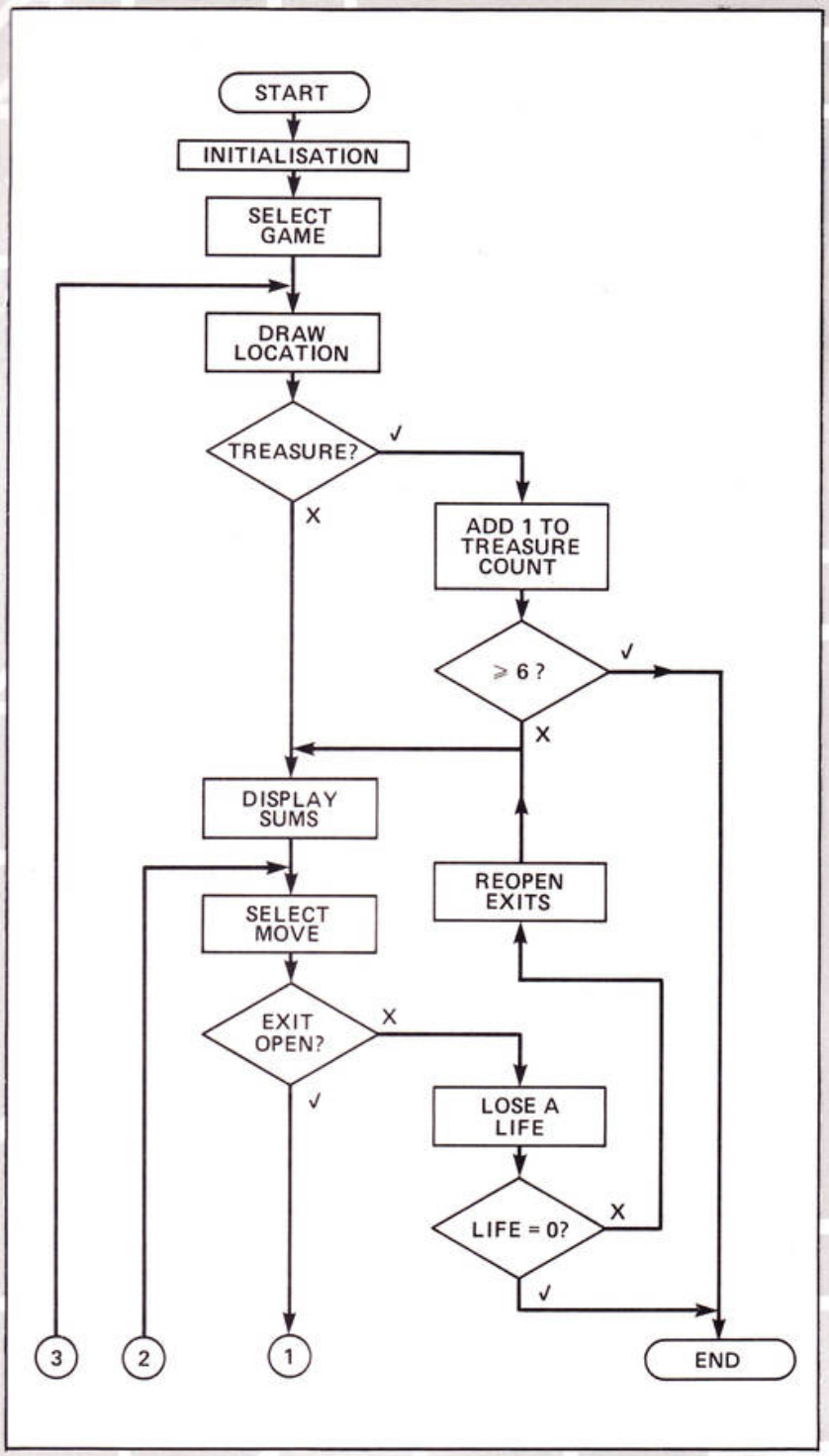
game type	if 1 then sums are addition if 2 then sums are subtraction if 3 then sums are multiplication
level	1 to 99, this determines the range of values each sum can take
a$	used for input from questions
q$(12)	array for data that specifies an old game. Used to save current game if it is desired to end the game before it is finished.
f,g	general loop variables
S$(704)	character array the size of the screen. Used to store what is on the screen
m(10,10)	the codes for the maze layout are stored in this array
1(11,4)	the 11 locations have a possible 4 exits. This array contains a binary representation of the exits from a given room. 1(x,1) will have a 1 in it if there is an exit to the left and a 0 for no exit. 1(x,2) for down, 1(x,3) for right, 1(x,4) for up.
p(2)	the maze co-ordinates of the current location
e$	a string of 32 spaces used to rubout text on the screen
notreas	number of treasures found
lives	number of remaining lives
score	current score

MATHS MAZE



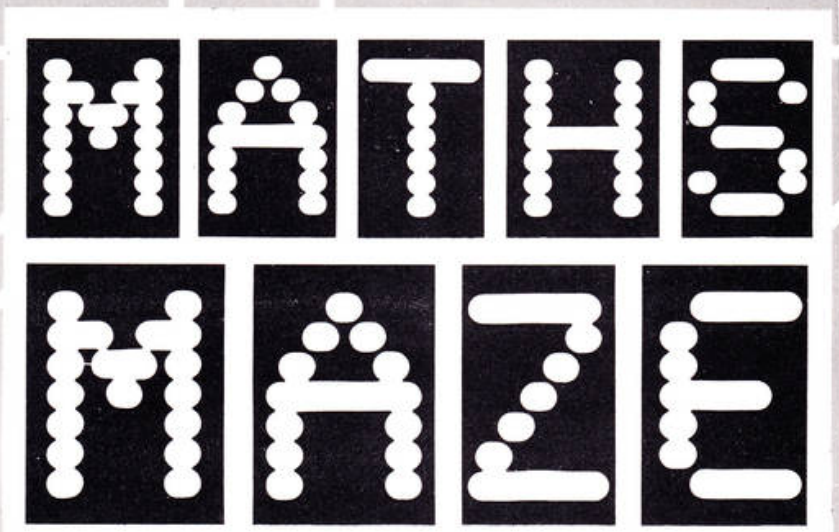
t(12)	stores the locations of the six pieces of treasure; t(1), t(3) etc = x-coordinates, t(2), t(4) etc = y-coordinates
a	stores the code number of the current maze location. Used to save computing time caused by repeated evaluation of the expression m(p(1), p(2)) which is used in a number of calculations
wrong	a count of the number of incorrect responses to a sum. Two chances are given to get it right
treasure	if 1 there is treasure in the room if 0 there is no treasure
ans	attemped answer to the sum
sumtotal	correct answer
q(4)	an array for storing the correct answers to all the sums at the exits to a location
sum 1, sum 2	two parts that make up each sum
xman1,xman2,yman	co-ordinates of the man's location. Used to rub him out when an exit has been blocked
i$,j$	graphic chars to define man. Used to make him walk
limit	the number of co-ordinates to be read to define a location

```
  5 GO SUB 4500
  9 REM the routine at 8000
sets up the static maze features
 10 GO SUB 8000
 19 REM the routine at 8270
sets up the maze for a new game
 20 GO SUB 8270
 30 GO SUB 4000
 40 GO SUB 7000
 50 GO SUB 6000
 60 GO TO 40
3999 REM select level
4000 PAPER 5: BORDER 1: INK 0: C
LS
4005 GO SUB 4200: REM ask if new
 game
4010 CLS : PRINT AT 6,5;"SELECT
GAME TYPE";AT 10,2;"Press  1  fo
r addition";AT 12,2;"Press  2  f
or subtraction";AT 14,2;"Press
3  for multiplication"
4015 BEEP .3,10
4020 IF INKEY$="" THEN  GO TO 40
20
4030 IF INKEY$<>"1" AND INKEY$<>
"2" AND INKEY$<>"3" THEN  GO TO
4020
4040 LET gametype=VAL INKEY$
4050 CLS : PRINT AT 8,5;"SELECT
LEVEL  1 - 99";AT 15,6;"1 - Easy
, 99 - Hard"
4055 BEEP .3,10
4060 PRINT AT 20,5;"Type number
then ENTER"
4070 INPUT AT 0,14;level
4080 IF level<1 OR level>99 THEN
  GO TO 4070
4090 INK 0: PAPER 7: BORDER 0: C
LS
4100 RETURN
4199 REM load an old game ?
4200 BEEP .3,10: PRINT AT 10,5;"
Do you want to continue";AT 12,8
;"a previous game ?";AT 14,9;"(t
ype y or n)"
4210 LET a$=INKEY$
4220 IF a$="" THEN  GO TO 4210
4230 IF a$="n" OR a$="N" THEN  G
O TO 4490
4240 IF a$="y" OR a$="Y" THEN  G
O TO 4260
4250 GO TO 4210
4260 CLS : PRINT AT 10,5;"(T)ape
 or (M)icrodrive ?"
4270 LET a$=INKEY$
4280 IF a$="" THEN  GO TO 4270
4290 IF a$="T" OR a$="t" THEN  G
O TO 4320
4300 IF a$="M" OR a$="m" THEN  G
O TO 4350
4310 GO TO 4270
4315 REM tape load old game
4320 PAUSE 0: PRINT AT 15,4;"Pre
pare tape for LOADing";AT 19,5;"
Press any key to LOAD"
4330 PAUSE 0: LOAD "" DATA q$()
4340 GO TO 4370
4345 REM microdrive load old gam
e
4350 PAUSE 0: PRINT AT 15,2;"Pre
pare microdrive cartridge";AT 19
,5;"Press any key to LOAD": PAUS
E 0
4360 LOAD *"m";1;"maze" DATA q$(
)
4370 LET p(1)=VAL q$(1 TO 2): LE
T p(2)=VAL q$(3 TO 4): LET lives
=VAL q$(5): LET notreas=VAL q$(6
): LET score=VAL q$(7 TO 9): LET
 level=VAL q$(10 TO 11): LET gam
etype=VAL q$(12)
4380 FOR f=1 TO notreas*2 STEP 2
: LET t(f)=0: LET t(f+1)=0: NEXT
 f: REM set treasure already fou
nd to zero to stop them being fo
und again
4385 PAPER 7: INK 0: BORDER 0: C
LS
4390 LET s$(154)=q$(6): LET s$(3
14)=q$(5): LET s$(474 TO 476)=q$
```




```
(7 TO 9)
4400 GO TO 40
4490 RETURN
4499 REM initial display
4500 BORDER 4: INK 1: PAPER 4: C
LS
4510 PRINT AT 2,3;"▗▄▄▄▖ ▗▄▄▖ ▗▄
▄▄▖ ▐ ▐ ▐▛▀▀▘"
4520 PRINT AT 3,3;"▐ ▐ ▐ ▐ ▐
▐ ▐ ▐▌"
4525 PRINT AT 4,3;"▐ ▐ ▐ ▐▄▄▄
▐ ▐▄▄▟ ▐▄▄▟"
4530 PRINT AT 5,3;"▐ ▐ ▐ ▐ ▐
▐ ▐ ▐ ▐"
4535 PRINT AT 6,3;"▐ ▐ ▐ ▐ ▐
▐ ▐ ▐ ▐"
4540 PRINT AT 7,3;"▐ ▐ ▐ ▐ ▐
▐ ▐ ▐ ▐▄▄▄"
4550 PRINT AT 10,6;"▗▄▄▄▖ ▗▄▄▖
▀▀▜ ▐▀▀▀"
```

```
4555 PRINT AT 11,6;"▐ ▐ ▐ ▐ ▐
▐ ▐"
4560 PRINT AT 12,6;"▐ ▐ ▐ ▐▄▄▄
▐ ▐▄▄▄"
4565 PRINT AT 13,6;"▐ ▐ ▐ ▐ ▐
▐ ▐"
4570 PRINT AT 14,6;"▐ ▐ ▐ ▐ ▐
▐ ▐"
4575 PRINT AT 15,6;"▐ ▐ ▐ ▐ ▐
▐▄▄▄ ▐▄▄▄"
4580 PRINT AT 19,6;"By J.S.Thurl
by1984"
4990 RETURN
5499 REM lose a life
5500 PAPER 5: INK 0: PRINT AT 19
,0;e$;AT 20,0;e$;AT 21,0;e$
5510 PRINT AT 19,1;"You must los
e a life to escape";AT 21,7;"fro
m this room"
5515 PAPER 7: INK 0: LET s$(300)
=" "
5520 LET lives=lives-1: LET s$(3
14)=STR$ lives
5530 IF lives=0 THEN GO TO 9800
: REM end of game-lost
5540 REM can still play
5545 REM calculate new sums
5550 PAUSE 100
5560 FOR f=1 TO 4: IF l(a,f)=-1
THEN LET l(a,f)=1
5570 NEXT f
5580 GO SUB 7140
5590 GO TO 6000
5999 REM select a move
6000 LET a=m(p(1),p(2))
6010 LET wrong=0
6015 IF treasure=1 THEN GO SUB
8500: LET treasure=0: REM pick u
p treasure
6020 LET a$=INKEY$: PRINT AT 16,
20; FLASH 1; BRIGHT 1; PAPER 2;
INK 7;"SELECT EXIT"
6030 IF a$="" THEN GO TO 6020
6032 PRINT AT 16,20;"
"
6035 REM move the man
6037 INK 3
6040 IF a$="5" AND l(a,1)=1 THEN
GO SUB 6600: GO TO 6100
6050 IF a$="6" AND l(a,2)=1 THEN
GO SUB 6700: GO TO 6100
6060 IF a$="8" AND l(a,3)=1 THEN
GO SUB 6800: GO TO 6100
6070 IF a$="7" AND l(a,4)=1 THEN
GO SUB 6900: GO TO 6100
6075 IF a$=" STOP " THEN GO TO
7500: REM abort game
6080 INK 0: FOR f=1 TO 20: BORDE
R 2: BEEP .01,-20: BORDER 7: NEX
T f: BORDER 0
6090 GO TO 6020
6100 INK 0: PRINT AT 19,0; INK 0
; PAPER 3;"What is the answer to
this sum? "
6110 PRINT AT 20,0; PAPER 3;e$
6120 PRINT AT 21,0; INK 0; PAPER
3;" Type the answer then ENTER
"
6125 BEEP .3,10
6130 INPUT ans
6140 IF ans=sumtotal THEN GO TO
6340: REM correct answer
6145 REM got it wrong
6150 LET score=score-3: IF score
<0 THEN LET score=0
6160 IF wrong=1 THEN GO TO 6240
6170 BEEP .3,2: BEEP .4,-5
6180 PAPER 3: PRINT AT 19,0;e$;A
T 20,0;e$;AT 21,0;e$: PAPER 7
6190 PRINT AT 20,8; INK 7; PAPER
2;"WRONG - TRY AGAIN"
6210 PRINT AT 14,25;" ";AT 14,
25;score: LET s$(474 TO 476)=STR
$ score: PAUSE 100: PRINT AT 20,
0;e$
6220 LET wrong=wrong+1
6230 GO TO 6100
6235 REM block the exit
6240 PAPER 4: PRINT AT 19,0;e$;A
T 20,0;e$;AT 21,0;e$
6250 PRINT AT 19,0;"SORRY - You
cannot go this way any m
ore";AT 21,2;"The correct answer
was ";sumtotal
6260 PAPER 7: INK 2: IF a$="5" T
HEN LET l(a,1)=-1: PRINT AT 10,
0;" ": FOR f=4 TO 13: PRIN
T AT f,3;"#": NEXT f
6270 IF a$="6" THEN LET l(a,2)=
-1: PRINT AT 14,4;"##########";A
T 16,5;" "
6280 IF a$="8" THEN LET l(a,3)=
-1: PRINT AT 6,11;" ": FOR
f=4 TO 13: PRINT AT f,14;"#": N
EXT f
6290 IF a$="7" THEN LET l(a,4)=
-1: PRINT AT 3,4;"##########";AT
1,5;" "
6291 INK 0
6295 PRINT AT xman1,yman;" ";AT
xman2,yman;" ": REM rubout old m
an position
```




```
6300 PRINT AT 8,9;"▗";AT 9,9;"▚"
: PAUSE 150
6305 IF l(a,1)<=0 AND l(a,2)<=0
AND l(a,3)<=0 AND l(a,4)<=0 THEN
  GO TO 5500: REM no way out of
location
6310 PAPER 1: INK 6: PRINT AT 19
,0;e$;AT 20,0;e$;AT 21,0;e$;AT 2
0,6;"TRY ANOTHER EXIT"
6315 PAPER 7: INK 0: PAUSE 100:
FOR f=19 TO 21: PRINT AT f,0;e$:
 NEXT f: PRINT AT 20,0;s$(642 TO
 671)
6320 GO TO 6000
6330 REM got it right
6340 INK 6: PAPER 2: PRINT AT 19
,0;e$;AT 20,0;e$;AT 21,0;e$;AT 2
0,7;"CORRECT - WELL DONE"
6350 INK 0: PAPER 7: BEEP .3,5:
BEEP .3,10
6360 REM move to new location
6370 IF a$="5" THEN  LET p(2)=p(
2)-1
6380 IF a$="6" THEN  LET p(1)=p(
1)+1
6390 IF a$="7" THEN  LET p(1)=p(
1)-1
6400 IF a$="8" THEN  LET p(2)=p(
2)+1
6410 LET score=score+1: LET s$(4
74 TO 476)=STR$ score
6420 REM clear location part of
s$
6430 FOR f=1 TO 18*32 STEP 32: L
ET s$(f TO f+17)="
    ": NEXT f
6440 FOR f=1 TO 4: IF l(a,f)<0 T
HEN  LET l(a,f)=1
6450 NEXT f
6590 RETURN
6599 REM move man left
6600 LET sumtotal=q(1)
6610 PAUSE 10: FOR f=9 TO 1 STEP
 -3
6620 PRINT AT 9,f+1;" ";AT 8,f+1
;" ";AT 9,f;"J";AT 8,f;"F": PAUS
E 17
6630 PRINT AT 9,f;" ";AT 8,f;" "
;AT 9,f-1;"J";AT 8,f-1;"L": PAUS
E 17
6640 PRINT AT 9,f-1;" ";AT 8,f-1
;" ";AT 9,f-2;"G";AT 8,f-2;"F":
PAUSE 17
6670 NEXT f
6680 LET xman1=8: LET yman=1: LE
T xman2=9
6690 RETURN
6699 REM move man down
6700 LET sumtotal=q(2)
6710 PAUSE 10: FOR f=9 TO 13
6710 PAUSE 10: FOR f=9 TO 13
6720 IF INT (f/2)<>f/2 THEN  LET
  i$="▚"
6730 IF INT (f/2)=f/2 THEN  LET
i$="▗"
6740 PRINT AT f-2,9;" "
6750 PRINT AT f-1,9;"▗";AT f,9;i
$
6760 PAUSE 10
6770 NEXT f
6780 LET xman1=13: LET yman=9: L
ET xman2=12
6790 RETURN
6799 REM move man right
6800 LET sumtotal=q(3)
6810 PAUSE 10: FOR f=9 TO 16 STE
P 3
6820 PRINT AT 9,f-1;" ";AT 8,f-1
;" ";AT 9,f;"H";AT 8,f;"H": PAUS
E 17
6830 PRINT AT 9,f;" ";AT 8,f;" "
;AT 9,f+1;"G";AT 8,f+1;"H": PAUS
E 17
6840 PRINT AT 9,f+1;" ";AT 8,f+1
;" ";AT 9,f+2;"I";AT 8,f+2;"H":
PAUSE 17
6870 NEXT f
6880 LET xman1=8: LET yman=16: L
ET xman2=9
6890 RETURN
6899 REM move man up
6900 LET sumtotal=q(4)
6910 PAUSE 10: FOR f=9 TO 5 STEP
 -1
6920 IF INT (f/2)<>f/2 THEN  LET
  i$="▚"
6930 IF INT (f/2)=f/2 THEN  LET
i$="▗"
6940 PRINT AT f+1,9;" "
6950 PRINT AT f,9;i$;AT f-1,9;"▗
"
6960 PAUSE 10
6970 NEXT f
6980 LET xman1=5: LET yman=9: LE
T xman2=4
6990 RETURN
6999 REM draw a location
7000 RESTORE 9000+m(p(1),p(2))*1
0
7010 READ limit
7020 FOR f=1 TO limit: READ a: L
ET s$(a)="■": NEXT f
7035 REM put allowed moves on sc
reen
7040 READ s$(642 TO 671)
7050 LET s$(266)="▗": REM top ha
lf of man
7060 LET s$(298)="▚": REM bottom
 half of man
7065 REM check for treasure
7070 LET treasure=0
7080 FOR f=1 TO 12 STEP 2
7082 IF p(1)=t(f) AND p(2)=t(f+1
) THEN  LET treasure=1: LET t(f)
=0: LET t(f+1)=0: GO TO 7090
7085 NEXT f
7090 IF treasure=0 THEN  GO TO 7
130
7095 REM found treasure
7100 LET s$(300)="△": REM treasu
re chest
7120 LET score=score+20: LET s$(
474 TO 476)=STR$ score: LET notr
eas=notreas+1: LET s$(154)=STR$
notreas
7130 IF notreas>=6 THEN  GO TO 9
700: REM wins game
7135 REM calculate sums
7140 RESTORE 9005+m(p(1),p(2))*1
0
7150 DIM q(4)
7160 FOR f=1 TO 4: IF l(m(p(1),p
(2)),f)<>1 THEN  GO TO 7260
7170 LET sum1=INT (RND*(level+10
))
7180 LET sum2=INT (RND*(level+10
))
7190 IF gametype=1 THEN  LET q(f
)=sum1+sum2: READ a,b: LET s$(a
TO b)=STR$ sum1+"+"+STR$ sum2: G
O TO 7260
7200 IF gametype=3 THEN  LET q(f
)=sum1*sum2: READ a,b: LET s$(a
TO b)=STR$ sum1+"*"+STR$ sum2: G
O TO 7260
 TO b)=STR$ sum1+"+"+STR$ sum2: G
O TO 7260
7200 IF gametype=3 THEN  LET q(f
)=sum1*sum2: READ a,b: LET s$(a
TO b)=STR$ sum1+"*"+STR$ sum2: G
O TO 7260
7210 READ a,b: LET q(f)=ABS (sum
1-sum2)
7220 IF sum1>=sum2 THEN  LET s$(
a TO b)=STR$ sum1+"-"+STR$ sum2
7230 IF sum1<sum2 THEN  LET s$(a
 TO b)=STR$ sum2+"-"+STR$ sum1
7240 NEXT f
7270 PRINT AT 0,0;s$
7280 RETURN
7499 REM abort game
7500 BORDER 5: PAPER 5: INK 0: C
LS
7510 PRINT AT 10,5;"Do you want
to SAVE the";AT 12,7;"current po
sitions ?";AT 14,9;"(type y or n
)"
7520 LET a$=INKEY$
7530 IF a$="" THEN  GO TO 7520
7540 IF a$="y" OR a$="Y" THEN  G
O TO 7570
7550 IF a$="n" OR a$="N" THEN  G
O TO 7670
7560 GO TO 7520
7565 REM save the game
7570 DIM q$(12): LET q$(1 TO 2)=
STR$ p(1): LET q$(3 TO 4)=STR$ p
(2): LET q$(5)=STR$ lives: LET q
$(6)=STR$ notreas: LET q$(7 TO 9
)=STR$ score: LET q$(10 TO 11)=S
TR$ level: LET q$(12)=STR$ gamet
ype
7575 CLS : PRINT AT 10,5;"(T)ape
 or (M)icrodrive ?"
7580 LET a$=INKEY$
7590 IF a$="" THEN  GO TO 7580
7600 IF a$="t" OR a$="T" THEN  G
O TO 7630
7610 IF a$="m" OR a$="M" THEN  G
O TO 7650
7620 GO TO 7580
7625 REM tape save routine
7630 SAVE "maze" DATA q$()
7640 GO TO 7670
7645 REM microdrive save routine
7650 PAUSE 0: PRINT AT 21,7;"Pre
ss a key to save": PAUSE 0
7660 ERASE "m";1;"maze": SAVE *"
m";1;"maze" DATA q$()
7665 REM carry on ?
7670 CLS : PRINT AT 10,5;"Do you
 want to play";AT 12,9;"some mor
e ?";AT 14,8;"(type y or n)"
7680 LET a$=INKEY$
7690 IF a$="y" OR a$="Y" THEN  B
ORDER 0: PAPER 7: CLS : GO TO 40
7700 IF a$="n" OR a$="N" THEN  N
EW
7710 GO TO 7680
7999 REM initialisation routine
8000 REM set up maze m(10,10)
8005 DIM m(10,10)
8010 RESTORE 9200: FOR f=1 TO 10
: FOR g=1 TO 10
8020 READ m(f,g)
8030 NEXT g: NEXT f
```

```
8120 REM set up UDG characters
8130 RESTORE 8160: FOR f=1 TO 17
8140 READ s$: FOR g=0 TO 7: READ
 a: POKE USR s$+g,a
8150 NEXT g: NEXT f
8160 DATA "A",0,60,60,60,24,126,
126,90,"B",90,90,24,36,36,36,102
,0,"C",0,24,36,60,66,66,126,0
8170 DATA "D",90,88,28,36,36,38,
96,0,"E",90,26,56,36,36,100,6,0
8175 DATA "F",8,8,12,12,12,12,28
,0,"G",16,16,48,48,48,48,56,0
8180 DATA "H",0,28,28,8,28,42,42
,76,"I",8,8,24,36,34,98,6,0,"J",
0,56,56,16,56,84,84,50
8190 DATA "K",16,16,24,36,68,70,
96,0,"L",8,8,8,20,19,17,48,0,"M"
,16,16,16,40,200,136,12,0
8195 DATA "N",129,189,189,189,15
3,255,24,24,"O",24,24,24,36,36,3
6,102,0
8200 DATA "P",0,28,28,8,28,42,42
,42,"Q",0,56,56,16,56,84,84,84
8205 REM set up move matrix for
each location
8210 REM l(x,1)=left move,l(x,2)
=down move,l(x,3)=right move,l(x
,4)=up move--if 1 then OK to go
that way,if 0 then no exit,if -1
 then exit blocked due to wrong
answer.
8220 DIM l(11,4)
8230 RESTORE 8240: FOR f=1 TO 11
: READ l(f,1),l(f,2),l(f,3),l(f,
4): NEXT f
8240 DATA 1,1,1,1,1,1,0,1,0,1,1,
1,1,0,1,1,1,1,1,0,1,0,0,1,0,0,1,
1,1,1,0,0,0,1,1,0,0,1,0,1,1,0,1,
0
8250 RETURN
8259 REM set up screen display i
n array s$
8270 DIM s$(704)
8280 LET s$(55 TO 61)="No. of":
LET s$(86 TO 95)="treasures": LE
T s$(154)="0"
8290 LET s$(246 TO 255)="Lives l
eft": LET s$(314)="3"
8300 LET s$(408 TO 413)="Score":
 LET s$(474)="0"
8310 LET s$(537 TO 609)="*******
*************************"
8320 FOR f=0 TO 18: LET s$(f*32+
19)="*": NEXT f
8330 REM current location is sto
red in p
8340 DIM p(2)
8350 LET p(1)=INT (RND*9)+1: LET
 p(2)=INT (RND*9)+1
8360 IF m(p(1),p(2))=0 THEN  GO
TO 8350
8395 REM e$ is used to clear a
line of text
8400 LET e$="
                "
8410 LET notreas=0: LET lives=3:
 LET score=0
8415 REM notreas=no. of treasure
s held by the player
8420 DIM t(12)
8425 REM t stores the locations
of the treasure - t(n)=xcoord,t(
n+1)=ycoord
8430 FOR f=1 TO 12 STEP 2
8440 LET t(f)=INT (RND*9)+1
8450 LET t(f+1)=INT (RND*9)+1
8460 NEXT f
8470 RETURN
8499 REM move man to pick up tre
asure
8500 FOR f=1 TO 8: BEEP .05,20:
BEEP .1,10: NEXT f
8510 FOR f=9 TO 11: PRINT AT 8,f
-1;" ";AT 9,f-1;" ";AT 8,f;"H";A
T 9,f;"K": NEXT f
8520 PAUSE 30: PRINT AT 8,11;"N"
;AT 9,11;"O";AT 7,11;"A": PAUSE
50
8530 PRINT AT 7,11;" ": FOR f=11
 TO 9 STEP -1: PRINT AT 8,f+1;"
";AT 9,f+1;" ";AT 8,f;"F";AT 9,f
;"J": NEXT f
8540 PAUSE 10: PRINT AT 8,9;"I";
AT 9,9;"K"
8550 PRINT AT 16,20; FLASH 1; BR
IGHT 1; PAPER 2; INK 7;"SELECT E
XIT"
8990 RETURN
8999 REM data for drawing the ma
ze locations
9000 REM data for design 1
9010 DATA 28,4,36,68,100,99,98,9
7,15,47,79,111,112,113,114,449,4
50,451,452,484,516,548,463,464,4
65,466,495,527,559
9012 DATA "5=left  6=down  7=up
  8=right"
9015 DATA 321,327,518,526,204,21
0,38,44
9019 REM data for design 2
9020 DATA 32,4,36,68,100,99,98,9
7,449,450,451,452,484,516,548,15
,47,79,111,143,175,207,239,271,3
03,335,367,399,431,463,495,527,5
59
9022 DATA "    5=left  6=down  7
=up"
9025 DATA 321,327,518,526,38,44
9029 REM data for design 3
9030 DATA 32,4,36,68,100,132,164
,196,228,260,292,324,356,388,420
,452,484,516,548,15,47,79,111,11
2,113,114,463,464,465,466,495,52
7,559
9032 DATA "    6=down  7=up   8=
right"
9035 DATA 518,526,204,210,38,44
9039 REM data for design 4
9040 DATA 32,4,36,68,100,99,98,9
7,15,47,79,111,112,113,114,449,4
50,451,452,453,454,455,456,457,4
58,459,460,461,462,463,464,465,4
66
9042 DATA "    5=left  7=up   8=
right"
9045 DATA 321,327,204,210,38,44
9049 REM data for design 5
9050 DATA 32,97,98,99,100,101,10
2,103,104,105,106,107,108,109,11
0,111,112,113,114,449,450,451,45
2,484,516,548,463,464,465,466,49
5,527,559
9052 DATA "    5=left  6=down  8
=right"
9055 DATA 321,327,518,526,204,21
0
9059 REM data for design 6
9060 DATA 36,4,36,68,100,99,98,9
7,449,450,451,452,453,454,455,45
6,457,458,459,460,461,462,463,15
,47,79,111,143,175,207,239,271,3
03,335,367,399,431
9062 DATA "       5=left      7=
up"
9065 DATA 321,327,38,44
9069 REM data for design 7
9070 DATA 36,4,36,68,100,132,164
,196,228,260,292,324,356,388,420
,15,47,79,111,112,113,114,452,45
3,454,455,456,457,458,459,460,46
1,462,463,464,465,466
9072 DATA "       7=up       8=ri
ght"
9075 DATA 204,210,38,44
9079 REM data for design 8
9080 DATA 36,97,98,99,100,101,10
2,103,104,105,106,107,108,109,11
0,111,449,450,451,452,484,516,54
8,143,175,207,239,271,303,335,36
7,399,431,463,495,527,559
9082 DATA "       5=left      6=
down"
9085 DATA 321,327,518,526
9089 REM data for design 9
9090 DATA 36,100,101,102,103,104
,105,106,107,108,109,110,111,112
,113,114,463,464,465,466,495,527
,559,132,164,196,228,260,292,324
,356,388,420,452,484,516,548
9092 DATA "       6=down      8=
right"
9095 DATA 518,526,204,210
9099 REM data for design 10
9100 DATA 36,4,36,68,100,132,164
,196,228,260,292,324,356,388,420
,452,484,516,548,15,47,79,111,14
3,175,207,239,271,303,335,367,39
9,431,463,495,527,559
9102 DATA "       6=down      7=
up"
9105 DATA 518,526,38,44
9109 REM data for design 11
9110 DATA 36,97,98,99,100,101,10
2,103,104,105,106,107,108,109,11
0,111,112,113,114,449,450,451,45
2,453,454,455,456,457,458,459,46
0,461,462,463,464,465,466
9112 DATA "       5=left      8=
right"
9115 DATA 321,327,204,210
9199 REM data for maze
9200 DATA 9,5,11,5,11,5,5,5,11,8
9210 DATA 10,7,5,1,11,2,3,4,11,2
9220 DATA 3,11,6,3,8,3,6,9,8,10
9230 DATA 3,5,5,4,4,1,11,1,1,2
9240 DATA 3,6,3,8,9,2,9,6,3,2
9250 DATA 3,5,2,7,6,3,2,9,6,10
9260 DATA 3,6,10,9,5,2,7,4,5,2
9270 DATA 10,9,2,7,2,3,5,5,6,10
9280 DATA 10,7,1,8,10,10,10,3,8,
10
9290 DATA 7,11,6,7,4,4,6,7,4,6
9499 REM end of game display
9500 BORDER 2: PAPER 2: INK 6: C
LS
9510 PRINT AT 0,2;"
 ▌        ▪
9520 PRINT AT 1,2;"
 ▌        ▟▀▀▀▪
9530 PRINT AT 2,2;"▛▀▀▜  ▛▀▀▜  ▛▀
▀▜  ▛▀▀▜  ▌  ▌"
9540 PRINT AT 3,2;"▙▄▄▟  ▌  ▌  ▌
 ▌  ▌  ▌  ▌▀▀▪"
9550 PRINT AT 4,2;"▌     ▌  ▌  ▌
 ▌  ▌  ▌  ▌▌"
9560 PRINT AT 5,2;"▙▄▄▟  ▌  ▌  ▌▙
▄▄  ▙▄▄▟  ▌"
9570 PRINT AT 8,5;"▛▀▀▜  ▛▀▀▜  ▛▀▜
▝▀▀▀▜  ▛▀▀▜"
9580 PRINT AT 9,5;"▌  ▌  ▌▛▀▀▜  ▌▌
▌  ▙▄▄▟"
9590 PRINT AT 10,5;"▌  ▌▌  ▌▌  ▌▌
 ▌▌"
9600 PRINT AT 11,5;"▙▄▄▟  ▙▄▄▟  ▌▌
 ▌  ▙▄▄▟"
9610 PRINT AT 12,8;"▌";AT 13,5;"
▝▀▀▀▀"
9690 RETURN
9699 REM game over - win
9700 GO SUB 9500
9710 PRINT AT 16,5;"YOU HAVE FOU
ND ALL";AT 18,8;"THE TREASURE"
9720 PAUSE 200: CLS : PRINT AT 1
0,5;"Do you want to play";AT 12,
8;"a new game ?";AT 14,8;"(type
y or n)"
9730 LET a$=INKEY$
9740 IF a$="" THEN  GO TO 9730
9750 IF a$="y" OR a$="Y" THEN  G
O TO 20
9760 IF a$="n" OR a$="N" THEN  N
EW
9770 GO TO 9730
9799 REM game over - lose
9800 GO SUB 9500
9810 PRINT AT 16,4;"YOU HAVE MAD
E TOO MANY";AT 18,10;"MISTAKES,"
;AT 20,4;"AND RUN OUT OF LIVES."
9820 GO TO 9720
9830 RETURN
9899 REM auto-run of game
9900 CLS : PRINT AT 5,5;"Type ""
t"" for TAPE";AT 7,10;"""m"" for
 MICRODRIVE"
9910 LET a$=INKEY$: IF a$="" THE
N  GO TO 9910
9920 IF a$="t" OR a$="T" THEN  G
O TO 9940
9930 IF a$="m" OR a$="M" THEN  G
O TO 9950
9935 GO TO 9910
9940 SAVE "mathsmaze" LINE 9960:
 GO TO 9960
9950 SAVE *"m";1;"mathsmaze" LIN
E 9960
9960 RUN
```

READERS

Readers give their views on some of the commercially available software that's around for the Sinclair micros.

SNOOKER
Vision Software
£6.95 Spectrum
Joystick option

This tape is a progression from the now 'Old Hat' Pool arcade and computer game. This version is written for 16 or 48K Spectrum.

This program has been written to follow the rules of snooker exactly. It is a 1 or 2 player game and as 1 player you have to try and better your score. There is on-screen scoring at all times and you are constantly reminded of how many shots and foul points you have accumulated so far.

After loading, the program automatically, checks to see if a Kempston Joystick is attached and if negative you can then use either of the three keyboard cursor controls ie, 8 directional keys around the S key or H key, or the usual cursor control keys with the S, H and g keys respectively to fire.

You are then given the choice of a 1 or 2 player game and whether you want a 15 ball game or not. You then place the white ball within the D on the table and play starts.

When cueing you not only have control over the power of your shot but you can also control the amount of side, bottom or top which is given to the cue ball which allows for a great amount of control for setting up your next shot. So after just 1 or 2 frames you will find that you are potting the balls in.

The precision and delicacy to place the customary cursor (+) sign is probably equal to playing a game on a full size table but with practice you learn which shots are possible. As the cursor tends to respond very quickly to the keyboard control I wonder if a slow cursor control would improve your game somewhat.

Sound effects are also included for both ball and cushion contract with another beep for a foul shot or for potting.

In all a more enjoyable game than Pool but that just might be due to my preference for a real game of snooker to that of Pool. However, this program will probably appeal to an adult more than a child due to the speed differences between snooker and Pool.

I am not disappointed in the price of £6.95 being spent and I am sure that if you purchase this game you will find that you return to it quite frequently.

STEVE CORTON

CHESS
Psion with Micro Gen
48K Spectrum
£7.95

This is an excellent version of the classic board game. The board and pieces are shown in high resolution graphics with the program being written almost entirely in machine code.

There are ten different levels of play although unfortunately there is no indication of which is the hardest. The cassette inlay boasts that level two will beat the user, and indeed the computer wins more often than not at that level.

Response time at the lower levels is about four seconds but at level nine it can take fifteen minutes between moves. Therefore I would advise people to play only at levels 0-2.

As with most chess programs on the market, the colours of the board and pieces can be changed to suit personal taste. I found red and cyan pieces on a black and white board were best.

The computer will recommend a move if you are stuck and more often than not I have found this a very useful asset.

The last twelve or so moves are displayed on the left of the screen, but unfortunately there is no way to save these except to write each move down.

There are also a number of other options available: you can save a game half way through and then play again when you feel like it, you can transfer the contents of the screen to an attached printer or you can set the board up to analyse problems and situations.

Altogether, for just £7.95 this program is surely one of the best programs available for the Spectrum.

TIMOTHY RICHARDS

VALHALLA
Legend
£14.95

The cassette was attractively boxed in a large black carton with a picture of a vikings head on it. Inside the cassette itself was housed in a plain cassette case. Included was a 48 page booklet of instructions which was well written and informative and as a plus was also entertaining. It is well worth reading the book first.

The cassette loaded first time every time and in fact is guaranteed to load or a replacement given. Loading takes about four minutes during which time you are presented with another picture of the vikings head in stunning graphics plus the program title. And it is well worth waiting the four minutes. The adventure gives graphics and text windows rather like "THE HOBBIT". Every location has a graphics display though some use the same display; these are beautiful and of course are in full colour.

The difference between these graphics and those of the "HOBBIT" is that this program shows all the characters, including yourself, and these also have movement. I have

included some sample screen dislays, though these fail to do justice in black and white without movement. These were obtained by the copy command, not available with "THE HOBBIT".

When fighting, characters are seen to strike at each other with swords of axes and on being given, say, the command "DRINK WINE" you can see the wine bottle tilted to the lips.

A feature I liked was that when killed, the game does not end, you go to hell and more or less have to start afresh.

It would be possible to spend many happy hours with this program just enjoying the superb graphics and trying out different commands. There are in actual fact six quests to carry out which must be completed in sequence. These all involve finding special objects like a word or a ring. The quests take a lot of thinking out and involve getting help from other characters who therefore need to be kept in with.

All the commands are easy to use and the program appears bug free (I have not yet crashed it). There appears to be many, many locations of which I have so far only visited twenty or thirty. There are thirty six main characters and twenty one different types of objects. Communication is very good and allows fairly sophisticted inputs such as "THOR SELL ME AXE FOR 20 CROWNS". Minor spelling errors are automatically corrected for you.

A save game facility is included which I always find useful on games of this length.

In conclusion I really enjoyed this game and I think it is streets ahead of the available opposition and although the price will be considered by some to be very high I think it is excellent value for money.

PAUL CARIS

GAMES DESIGNER
Quicksilva
£14.95

Once the games designer has loaded, a menu will appear listing options from 1-8. They are: 1) play game 2) select new game 3) alter sprites 4) configuration 5) movement 6) attack waves 7) lead 8) save to tape.

The play game option will allow you to play one of the eight games included or one of the games you have created yourself. The keys used to play the game can be changed to suit yourself.

Option 2 will allow you to select a game from 1-8. The game you choose can be played by choosing option 1 again. The eight games included are: attack of the mutant hamburgers, cyborg, reflection, turbo-spider, tanks-a-lot, halloween, splat and qbix.

When option 3 is chosen you are shown a number of pre-defined sprites or graphics which you can alter to suit your game. A simple sprite editor is included to help you redesign the sprites. The sprites used are made on a 12 by 12 grid. Each of the sprites can have a different colour. There are 30 sprites of which 15 are used for the aliens, 8 for the ships or laser bases and the rest are for the shields, missiles, and explosions.

If option 4 is selected you are shown another menu with options from 1-8. They are: 1) game format 2) background 3) foreground 4) special FX 5) missile sound 6) bomb sound 7) ship explode 8) alien explode.

The game format determines whether the game should be a scramble type game, an invaders type, an asteroid type or a berserk type game.

The special FX option lets you have star in the background and also if you should have a shield or not.

The rest of the options allow you to create sound effects for your games by moving certain ramps on the screen.

Lets return to option 5 of the first menu which is movement. You have to create patterns of movement that the aliens will follow.

When option 6 is chosen a chart appears with numbers from 0-7 on one side. These represent the 8 attack waves. The nice thing is that you can make each wave different from the others. You can make them faster and meaner or make more of them appear on the screen. It's all up to you!

The remaining two options will let you save the game you have made and lead it back again.

The games produced are very good and some may rival commercial software tapes. I am very sure you will have lots of fun creating your own games. A very good manual is also included and it goes through everything in detail.

FARLEY SCOLLICK

DEATH CHASE
Micromega
£6.95

The first time I tried playing this game was in the shop where I bought it, (the only time I've played it using a joystick). Anyway, the program loaded an excellent display of a man skidding round a corner on a

READERS REVIEWS

motorbike. "Looks good", I thought.

After a total loading time of about a minute, the computer started making the usual "beeping" noises ie similar to those at the beginning of Hungry Horace.

The screen for the game itself was split into two. The bottom half being green, and the top half being either blue (cyan), or black according to whether it was day or night, with all the intermediate colours ie dark blue, red etc, between patrols. In the middle of the bottom of the screen is the handlebars of a motorbike and two arms. On the horizon are a couple of trees.

Before I go any further I will tell you the story so far. "It is 2501, 100 years after "The Great War". The North American continent is ruled by mighty warlords in constant conflict over forest territory. You are one of the elite mercenaries, Riders of the Big Bikes. It's a quick way to get rich — and a quicker way to die. " Surprise, surprise!" You patrol your forest day and night, chasing enemy Riders and destroying them with your guided photon bolts for $1000 a time. You may find helicoptors and tanks too — your masters reward you particularly well if you destroy these."

Anyway, enough of the babble, how about the game itself?

You can use either a Kempston joystick or the keyboard to play. However, there are only three basic commands while playing the game, left, right and fire.

There is also accelerate and decelerate but I only ever used accelerate, and that only at the beginning of each patrol. As you start the game, the computer makes an attempt at imitating the sound of a motorbike and the trees start moving towards you in 3D. There are also two other motorbikes and sometimes a helicopter which flies from left to right corss the horizon or sometimes a tank. The aim of the game is to get through 8 sectors of forest, doing a day and night patrol in each. The game gets progressively harder as you go further on by adding more and more trees. In each patrol you must shoot the two motorbikes. However, you can only fire when at top speed and you have to wait about twenty seconds going at full speed and avoiding trees before they come into range. Shooting tanks and helicopters gets you bonus money.

As the literature itself says, "The greatest reward is for the Rider who can penetrate 8 sectors — you will need every ounce of skill to find out". Obviously I haven't enough ounces of skill, because although I have had the program about a month now, (and constantly play it), I still haven't finished the 8th sector.

On the whole I found the program almost terrifyingly realistic even if the story line wasn't so good.

It was quite reasonably priced for such a good program at £6.95. The screen graphics were excellent and very smooth. The instructions for loading were so simple that anyone could use them.

I would say that Death Chase is a must for anyone wanting an extremely realistic 3D game which is also a challenge and good value for money!

MICHAEL CLUBBE

REVERSI
Mine of Information
16K Spectrum
£7.95

The thing that struck me about this program was the amazing range of options available to the user.

The best of these was undoubtedly the action replay mode, during which you can go right back through a game to see where you made mistakes. This option appears to be unique to Mol and it certainly is a marvellous piece of programming.

The program is written in machine code and, as with the chess program above, it responds instantaneously in the lower levels. At level one (novice), I have yet to beat the computer. It really does play a mean game.

The graphics are nothing to write home about but with a game like this it is difficult to see how you can have excellent graphics.

Illegal moves are rejected so you can't cheat. This probably means I shall never beat the algorithm, but for £7.95 this program is excellent value for money.

TIMOTHY RICHARDS

ROCKETMAN
Software Farm
16K ZX81
£5.95

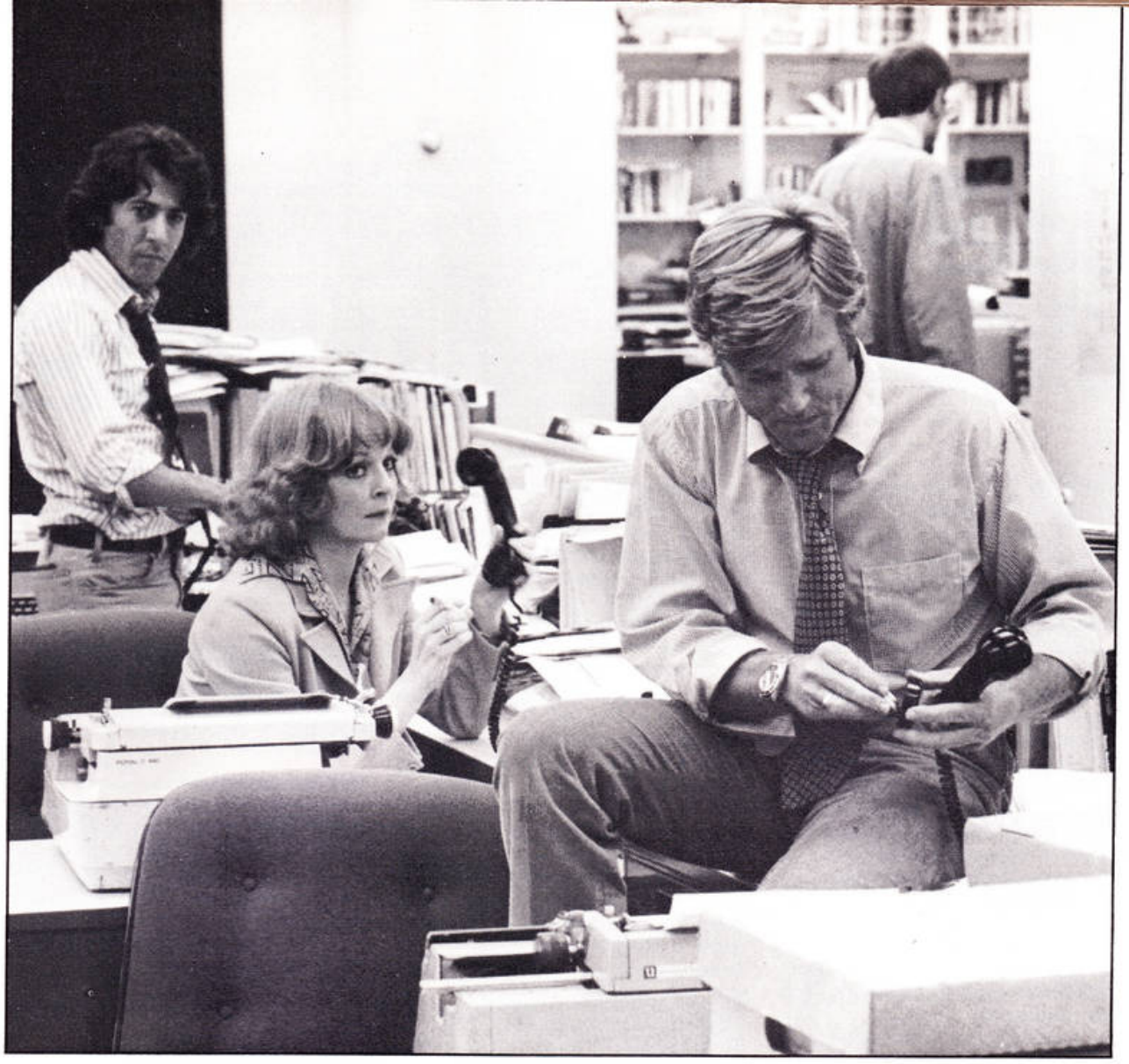
I have recently bought the new Hi-Res game "ROCKETMAN" from the Software Farm. It is their second Hi-Res game and is even better than "Forty Niner".

The object of the game is to collect powerpacks (1) on the platforms moving "Rocketman" (2) about using the ladders and when your fuel guage (3) is full you jump to the rocket (4), collect your jetpack and then fly over the left hand side of the screen where the diamonds (5) are waiting for you.

But things are not so easy as firstly there is a monster (7), emerging from the sea (6), called a "Bubloid" trying to catch you and drag you back into the sea and secondly if you miss a platform or ladder you fall into the sea anyway.

Layouts, speed and points change with each of the 6 different stages with the added twist that for the stages 4 to 6 the powerpacks and jetpack are replaced by legs of lamb and a vulture.

The game is fast and the graphics, as expected from the Software Farm, are superb and



free flowing.

If "Forty Niner" rates 9 out of 10 then "Rocketman" deserves 10 out of 10.

Prices £5.95 from the Software Farm, Freepost (BS 3658), Bristol BS8 2YY.

STUART BAILEY

SPECIAL AGENT
Fiveways
£9.95

Special Agent comes with a book which says this cassette is for 8-12 year olds but is playable by anyone. It is an educational package but fun to play.

The story is you are a special agent for M16, based in London. Your mission is to catch a dangerous enemy spy. To make your life a bit easier (thank goodness) you have an agent in every city, but... if your agent is spotted the spy will kill him! Your agents send you "intelligence" reports (some not so intelligent)!! Eg a report may be "Spy observed in capital city of USSR.." So you may fly to Moscow or lie in wait at Minsk. This brings us nicely into the travelling part of the game. We look at the map which fills the whole screen and decide where to go. So, press (I) and look at the timetable for both rail and air. The journeys are to different cities at any time in the day (and night)! You never seem to bother about getting on trains at 03.00, so at the bottom of the screen we see the message "WAITING TO TRAVEL" and (eventually) "TRAVELLING". We arrive and more often than not see the cheery message your agent here is alive and, no spy!! Although if your agent has been "knocked off" then you can hire a new one (in under a minute, isn't technology brilliant nowadays).

This is for the loss of £200!! Some of the intelligence reports are in code. It helps if you can work out the code quickly. At every city we go to our agent there tells us one or two letters. (Why can't they tell us all of it)? You also have the choice of looking at other cities timetables (to see where the spy may go next). In the book it gives you a few lines and a picture (Wow) about every capital city. You also have a map of Europe to help you in the book.

On the screen it tells you how much cash you've used (usually about £1000 for me)!! It also tells us how many days we've taken and the time of day (in 24-hour) time. You can pause the game to work out messages. If you are totally stuck you can quit the game!

Fiveways are really kind and give us a keyboard overlay. The BRITISH SECRET SERVICE are always short of money (ah)! (so spend as little as possible).

SPECIAL AGENT is priced £9.95 (and worth it) and available from Fiveways Software.

CHRIS LEACH

BETA BASIC
Betasoft
16K & 48K Spectrum
£11

This is a very comprehensive piece of software comprising a large number of 'utilities' for the

READERS REVIEWS

Spectrum. The tape provides two copies of the program for both the 16K and the 48K machine. It gave no problem at all in loading my 48K Spectrum.

A 32 page booklet accompanies the tape and sets out very clearly each function or Keyword and its applications often giving a short program as an example. Programs containing Beta BASIC can be SAVED to tape in the usual way. Before reLOADING, however, it is necessary to LOAD the Beta BASIC itself although if you do forget it is possible to MERGE after your own program has been LOADED.

In use the Beta BASIC is LOADED before starting programming so that the 27 new KEYWORDS and 10 new Functions are available.

Amongst the Keywords are the very useful and time saving AUTO number which provides an automatic line numbering facility with a step to your choice (greater than 10 but not greater than the line number — not really likely to present much of a problem). This works admirably and speeds up the entry of programs enormously unless of course they happen to be one of those irritating ones with odd, untidied line numbers — no excuses for these since there are plenty of renumbering programs available — a very fast one is included in this selection. The RENUM Keyword calls up a routine by which the whole Program, or selected blocks of it, is renumbered with your chosen step. One feature of many of the Keywords available in Beta BASIC is that they may be applied string-sliced, ie using (TO) to define their block of application, or not, when specific conditions then apply eg present line (indicated by the flashing curosr — another feature of this program) and a step of 10.

DELETE is included too to remove unwanted parts of your proram and can save annoying 'line No + ENTER' activity.

A frequent comment on Spectrum BASIC is that it is limited to IF — THEN statements and does not support DO WHILE or DO UNTIL. These are both provided in the Beta BASIC as is also EXIT IF, which provides the ability to leave a program from somewhere in the middle.

An interesting inclusion is the provision of CLOCK, providing a digital clock which will operate with or without a continuous display on the screen. This has also the facility to initiate either an audible alarm or to call up a subroutine or both.

Useful and unusual Keywords are provided to help the graphics designer. PLOT can be used, in the same way as its Spectrum BASIC equipment, but with the ability to position a string anywhere on the screen; ROLL gives a movement of all or part of the screen up, down, left or right with the items moved appearing on the opposite side of the screen ie nothing is destroyed. SCROLL on the other hand moves all or part of the screen in a similar way but that portion of the screen is lost. Both of these are illustrated by two short programs of which the ROLL is quite fascinating! Of the remaining Keywords SORT is a program for arranging strings or numbers in alphabetical or reverse order. This is very fast and can sort 100 ten-letter strings in about 0.2 sec.

If you have problems in debugging a program or deciding how it works then TRACE is yet another useful item included in this set of facilities.

Some 10 Functions are also available and these and the other Keywords not yet mentioned such as; PROC, END PROC, EDIT, LOOP, ON ERROR are all listed alphabetically in the booklet which also includes a list of the eight REPORTS which could occur in addition to those in Spectrum BASIC.

In all Beta BASIC is a very useful compendium of 'utilities' and although at first sight its price of £11 (inc. p&p) seems on the high side it is very little for each item included. Definitely recommended.

MICHAEL JACKMAN

COLOSSAL ADVENTURE
Level 9 Computing
48K Spectrum
£9.90

This is a complete version of the orginal mainframe "Adventure" game, for the 48K Spectrum. It is made up of two parts; a small BASIC program and about 30K of machine code which does all the work.

The cassette is accompanied by a small booklet, which describes the background to the game and contains instructions for loading and playing. The idea of the game is basically to find the Colossal Cavern and return with its treasure. This is very much easier said than done, because the game contains more than 200 locations and many different objects and creatures. Each location is individually described, and as special data compression techniques have been used the descriptions are rich in detail. The computer is instructed using short English phrases such as "move east" and "take bottle". Only a few of the possible commands are listed in the booklet, so part of the fun of the game is working out how to phrase your instructions to the program. Many words may be abbreviated.

The program responds very quickly to your instructions, which is just as well because I think it will take a long time to explore the entire scenario. It is also possible to save the current state of play onto tape.

Even if you do manage to recover the treasure, play continues, because Level 9 has added a completely new endgame with 70 extra locations. This makes the program

excellent value.

During your search for the treasure, you will come across several problems which must be solved before you can continue. Some of these puzzles are VERY difficult, so it is comforting that you also receive a coupon which entitles you to send off for one free clue.

Overall, I recommend this program very highly. The fact that it is a text-only adventure may put some people off, but in my opinion the detailed descriptions of locations are more realistic than graphical representations could ever be. Even the most hardened adventurer will have problems solving it, and it is a bargain at £9.90.

Colossal Adventure is available from LEVEL 9 COMPUTING, 229 Hughenden Road, High Wycombe, Bucks. HP13 5PG.

PAUL WRAY

THE DUNGEON MASTER
Crystal Software
£7.50

In this 'Dungeons and Dragons' game you play the game of the adventurer, whose quest is to find 10 Turquoise Rings before being killed. Of course there are the nasties out to get you.

Loading the game is easy and it loads first time, although this game suffers from a long loading time.

When loaded, you're invited to type in the name that you want to be known by in the dungeon — this can be up to 10 letters long. You are then asked to say which dungeon you want to play in (Dungeon 1 or 2). You are then given precise instructions on how to load the dungeon you have chosen.

After you have loaded the dungeon you are reminded of your quest. On pressing a key, you are shown your status. It looks something like this:

Strength	(Varies from 3 to 18)
Intelligence	(Varies from 3 to 18)
Wisdom	(Varies from 3 to 18)
Agility	(Varies from 3 to 18)
Charisma	(Varies from 3 to 18)
Wealth	(Varies from 3 to 18)

Strength is your muscular power, intelligence your ability at picking locks, and using magic items, wisdom is your ability at finding a secret door. Agility is your speed and charisma is the measurement of influence you can exert over others. When you kill monsters you gain experience points depending on how strong they were. When you reach certain amounts of experience, you go up levels. Gaining hit-points as you do so. Health is used up in changing levels. Your hit points are the amount of damage you can take.

In the game, your first battle is against a Kobold, an animal which is quite easy to kill. If you win, you will be able to take any useful objects in the room. The object is obtained by typing KEEP, which will provide an object then the number beside it and pressing ENTER. If you keep a suit of armour then the computer will automatically drop your old set. The easiest monster to kill is the Giant Centipede and the hardest is Demegorgon, a devil. Any single thing should usually be avoided, especially if it sounds nasty.

Also on this tape is a unique game called the dungeon creator and it gives the user practice in building up, extending, modifying and playing his or her dungeon. On loading this game you are asked if you want to save, build, modify, extend or play your dungeon. This loads every time, and is an extremely good function.

On the whole, this tape is well worth is price of £7.50 but I would not recommend it to people who have absolutely no idea of how to play D & D because you will end up attacking a lot of monsters that will kill you. Otherwise all fans of the cult will really enjoy this superb game.

MARTIN ROBB

GERMAN VOCABULARY TEST
Tutorial Software
Spectrum
£4.95

This is a user-friendly educational tape for the 48K ZX Spectrum. Each side is loaded by sensible commands. After stopping the tape as instructed, the user is treated to a graphics display of a 'German flag' being raised, followed by a sequence of beeps representing the German National Anthem.

Side A tests the user on nouns. A choice of ten sections, each containing fifteen nouns, is

READERS REVIEWS

given and the user selects one by entering a letter from 'a' to 'j'. The nouns in each section are associated with a particular subject. For example, one deals with "animals" whilst another is entitled "school".

Having made a choice, the user is presented in turn with five English nouns and is asked to enter 'der', 'die' or 'das' to indicate the gender and then to enter the German noun in each case. Entries of the genders and nouns are accompanied by audio signals. The user is also asked whether or not the 'umlaut' is required for certain vowels. When all five answers have been entered they are marked, half a point being awarded each for the correct gender and for the correctly spelt noun. Wrong answers are emphasized by a raspberry sound effect, and a score out of five is given.

Next the user's mistakes are corrected by an impressive sequence. First the correct noun is displayed, then it is spelt letter by letter on another part of the screen, after which the incorrect entry is displayed. Second the screen display changes and the user is asked to enter the correct noun. Incorrect entries are noted at this stage, and the foregoing sequence is repeated until the correctly spelt noun is entered. Also, the user is told the correct gender if this entry was wrong.

Having finished one section the user is invited to repeat it, or to choose another section. In either case the format is as described above, and at the end of each test a running score is displayed. Side B tests the user on verbs, adverbs and adjectives in the same style as for the nouns.

I found the correction sequence to be very helpful. Not knowing much German. I made plenty of mistakes and my scores were terrible. However, I soon learnt words and the scores improved with repeated use of the tape. My son studies O-level German and he soon got regular high scores (often 100%). The conclusion here is that this tape is good for a beginner, or perhaps for C.S.E revision. It would be excellent in parallel use with a grammar construction tape. In any case it is good value for money at £4.95.

German Vocabulary Test is available from Mr. I.M. Scott, Tutorial Software, "Vilands", Glasllwch Lane, Newport, Gwent, NPT 3PS.

E.W.J. MICHELL

ESCAPE
New Generation Software
16K Spectrum
£4.95

Once upon a time, in a distant valley, cut off from civilisation as we know it there was a maze. . . .

This is the title page to New Generation Software's "Escape" and it is from this maze that you must do just that.

Simple you might think, but you'd be wrong. Besides having to evade several species of dinasaur you also have to find an axe with which to break down the exit door.

You can choose from 5 levels of difficult each level relating to the number of dinosaurs. The first is easy because you only have one dinosaur to contend with. You can move twice as fast as the monsters, using the 5-8 keys, unless you are carrying the axe which slows you down because it is heavy. The axe is placed at random in the maze and when you walk (or should I say run) over it a message flashes up telling the axe using the 0 key — as I said before this slows you down considerably, but if you feel the hot breath of a hungry dinosaur on the back of your neck you can drop it by presing the 0 key again.

Level five is almost impossible because not only do you have to face earth-bound monsters but an airborne pteranodon which swoops down, and a Tricerotops that is small enough to be obscured by the hedges that make up the maze walls.

When, and if, you escape from the maze you are told the number of seconds you lasted and if you were good enough you can enter the Escape hall of fame. This can be saved for prosperity of you wish.

Full use is made of the Spectrum's colour, each monster being different. You can hear the patter of your own feet and those of the monsters through the sound facility and the graphics are excellent with a capital E. The monsters appear to walk most realistically and the maze is viewed from an elevated angle made possible by the use of the Spectrums "Bright" facility. This makes the top of the maze hedges a light green, while the sides are a darker shade of the same colour.

Escape is a very original and addictive game for the 16K Spectrum and one of the best that I have seen. It is well worth £4.95 and is widely available through major retailers such as W.H. Smith, or direct from New Generation Software, Oldland Common, Bristol.

MARTIN FLEGG

JOHNNY REB
M.C. Lothlorien
£5.50
48K Spectrum

This is another offering from the seemingly prolific 'War Game' situation software publishers, Lothlorien. This particular piece is written for a 48K Spectrum.

The game is set at a river crossing during the American Civil War. The screen representation shows a battlefield dissected by a river, which is randomly positioned each time the game is played. The bridge crossing the river is always approximately centre of screen.

Also on the screen is marshland and dense woodland, both of which you are unable to enter.

The object of the game is to either capture your opponent's flag, or to attain a major tactical advantage over your enemy. Either of these objectives should be reached within the time limit that you have set yourself.

Once the program has loaded you are asked if you want a 1 or 2 player game. The instructions for both are very similar. You are then given the choice of playing as the Union or the Confederates. For some reason I cannot bring myself to choose the Confederacy (perhaps because I believe in History repeating itself). But it doesn't seem to make any difference as the computer tends to get into a very strong position early in the game.

As previously mentioned, the game can have a time limit set, which is counted in game turns or you can also play until the conclusion and the enemy flag captured. The comprehensive instructions (in fact a small folding booklet) give advice from setting up play to conclusion and make learning to play a rapid process.

You have choice over the number of troops and their type ie. Infantry, Cavalry or Artillery. Your opponent obviously has the same as yourself, unless you play a handicap game. If your Spectrum is the opponent you can either have the machine with similar forces or with equal total power to yourself, with troops of its own choice.

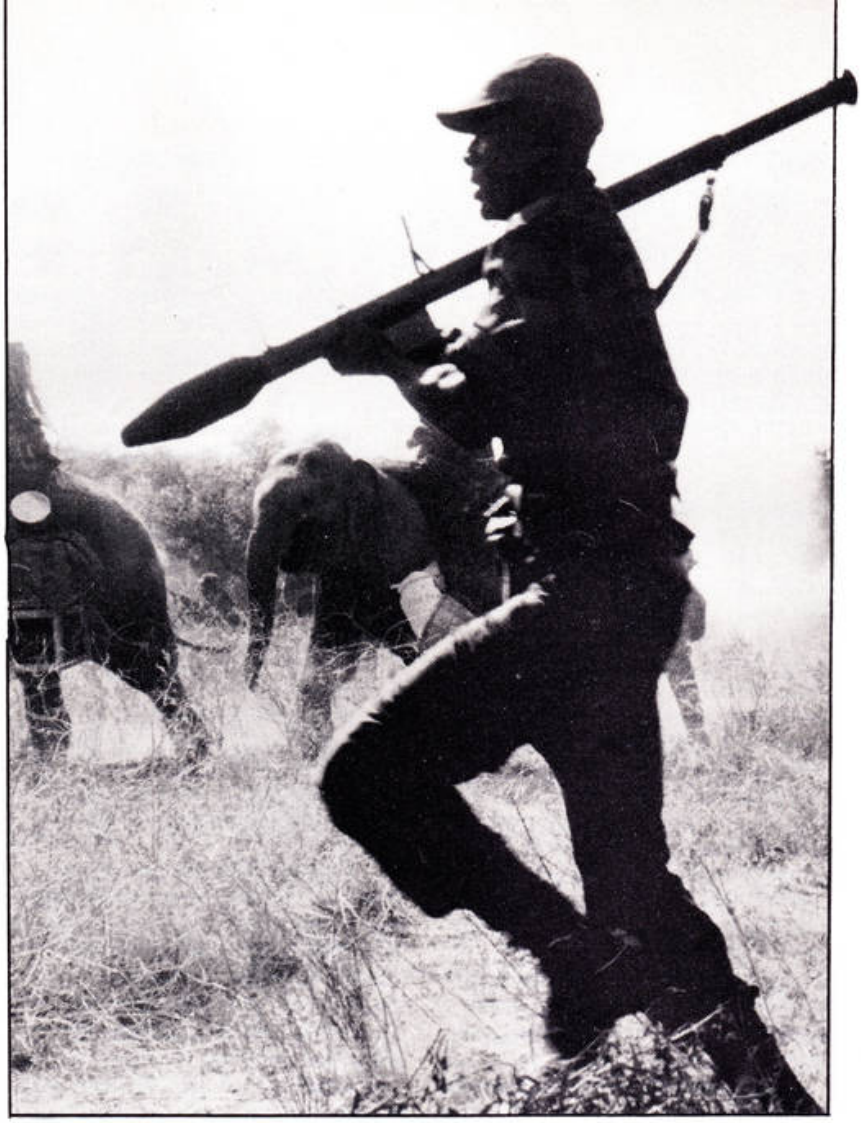
Once your troops have been chosen they will be randomly placed on the field for both sides. You may then move your pieces around in sequence, with player 1 always first. You may move in any of 8 compass directions up to a maximum of squares equivalent to the power of each unit. As the units are attacked by the enemy or the enemy artillery then their power reduces. This is also signified by a change of colour in the background square of each unit.

Once all your units have been moved or have fired then it is the opponent's turn, at the end of which the computer will check the strength of each unit. Play will continue with the next game turn. If you find that the evening has gone too quickly you have the option to 'Save Game' for continuation at a later date.

The game is very absorbing and I feel sure that your tactical strengths will be tested to the extreme whether you are playing the machine or a mere human opponent.

A well spent £5.50 on a well planned package, which also makes me want to find out more about the other programs from this stable.

STEVE CORTON



BACKGAMMON
CP Software
48K Spectrum

This program implements all the features of the board game with the board and pieces displayed in high resolution graphics.

The cassette inlay provides a brief summary of the rules of the game which was invaluable to someone like who had never played the game before. However, you soon get the hang of the game and it becomes quite addictive.

The program plays at one level only, so you either master it or you don't. The computer checks for illegal moves to prevent you from cheating. A recommend option would have been invaluable, but there is none.

The program is written in BASIC cum machine coded, with a response time of about three seconds.

An annoying fact is the ability of the computer to throw mammoth doubles just when it needs them and for the user to get low scores when he desperately requires high ones. This happened too many times to be coincidence so I would assume that somewhere in the program foul play is at work.

For £5.95 this is a fair representation of the real thing, although there are a couple of improvements that could be made.

TIMOTHY RICHARDS

EDITOR/ASSEMBLER
Picturesque
Spectrum 16K or 48K
£8.50

After messing around with machine-code for a few months I decided to buy an assembler program. An assembler/editor is a program, which will allow you to enter and edit mnemonics and label names and then it will

READERS REVIEWS

translate it into machine-code.

I decided to buy the EDITOR/ASSEMBLER program from PICTURESQUE. From the day I sent my order to the day the program bumped into my mailbox, two long weeks passed. Because the program now is compatible with the microdrives and because of some new facilities, the owner's manual are in two booklets. Reading the manuals from one end to another a few times gives you an idea of what the program can do, and which commands to use. You will have to jump from one manual to the other if you want to read it straight forward. PICTURESQUE will hopefully mix the two manuals together into one in the nearest future.

The cassette contains the 16K version on one side and 48K on the other. When run, the program asks which printer you use, if you want to save it or not (and I will recommend you to save it on another tape and use that, when you are going to use the program, and spare the original) and finally it asks you if it is a new program you'll produce or you want to continue with the one already stored in the SPECTRUM.

The program has AUTOmatic line numbers with start and step as you want, and it has RENUMBER. If you have to EDIT something, just type EDIT and line number, then you get that line down for editing. You can LIST in almost the usual way.

At the very beginning of a program you have to have an ORG command. ORG address gives the start address of the machine-code program. If you type ORG # the program will assemble in the right place — normally at 24014 — and you can run it from that point, inside the assembler program (a good way to test a new program).

You get a whole new character-set enabling you to have 40 characters in a line. They have a very good size — not too small.

Numbers may be in HEX or Decimal.

You can SAVE and VERIFY the text and/or the machine-code, and you can LOAD a saved text back into the program.

You cannot load a BASIC or a machine-code program into the assembler.

The assembling of a program is very fast. You can get the code printed on screen/paper — that will slow down the assembling a bit though.

Pointing out things I don't like I will start with the fact that the code only is printed in HEX. If you don't start the machine-code program from the beginning, you will have to translate the HEX address to decimal address.

Another thing I will point out, is that you cannot use graphic-characters in a message-string.

Generally I think this is superb program and indeed very user-friendly (except the HEX-code). I have already made some good machine-code subroutines to add to my BASIC programs and learned very much how the Spectrum works. I will highy recommend this program to everyone.

The EDITOR/ASSEMBLER costs £8.50 inc. VAT and p&p.

SOREN HANSEN

SUPERCHESS
C.P. Software
£4.95

The program loaded first time after about seven minutes (loading time). The game has ten levels and plays a surprisingly strong game even at the lower levels, and does not fall into traps easily. The program uses the standard system of algebraic notation and will not accept illegal moves. It caters for, as you would expect, castling and en-passant capture. Now for the game.

The program has fast response time on all the lower levels, the first two (0 and 1) normally responding within 10 seconds. However it is quite slow once above level 3 and extremely slow at level 9. The board display is clear and lists at the side of the board a list of the last 15 moves, what level you are playing at, and which of the two move evaluation modes you are in.

The game offers several options, "Help" which displays the option list and "Recommended Move" which offers you a good move, should you become stuck. "Self Play" gives you a demo game, the computer plays chess against itself. "Resign" allows you to resign mid game. "Analyse" allows you to change or set up the board into a particular position.

The game is surprisingly quick given the fact that the game is almost entirely written in BASIC. There are a few points however about this program, that need modification, I feel. Firstly, you cannot save a game to continue later and the analyse mode is rather slow and inconvenient. Also the game does not inform you when you are in check, in case you have not noticed this.

To summarise, the game is excellent value at £4.95 which is extremely cheap for a chess program. It is fairly fast and powerful and offers a challenge to virtually any player whatever his skill. However, the game would be even better if some of the faults were wiped out. The options, especially the recommended move would, I feel improve a beginner's game if he played the game enough times.

MICHAEL DIACK

ZX CHESS II
Artic
16K ZX81
£9.95

There is now a series of chess programs for the ZX81 on the market, some, which are

cheaper, aimed at the less serious players, and the others, like ZX Chess II (selling for £9.95) are aimed at already competent players who wish to improve their game. The package comes complete with seven playing abilities and many other options.

The instructions on the reverse side of the cover give a reasonable account of how to use the program.

Upon loading, you are asked the question 'PLAY, ANALYSE, OR LOAD'. If your answer is 'L' for load, the machine will load a previously saved game. Both in LOADing and SAVEing a very efficient process takes over — no program names are needed — and the actual SAVEing takes about two seconds. The 'Analysis' mode gives you the opportunity tyo solve a magazine chess puzzle, or to continue with a game of correspondence chess.

When the play begins the program has a number of sets of opening moves which contribute to making the game more random — a good thing. The board is set out so the black pieces spread across the top of the screen, and the white at the bottom. The squares are indicated by using the standard chess notation system. You are given a choice as to whether you want to be black or white, but I felt myself unable to play black, simply because I felt as if I were playing upside down (the pawns moved down the screen instead of going up). The computer allows every move, including en passent. When a faulty command is entered the program recognises the mistake, and prints 'INVALID' — a sign for you to try again. Pawns are automatically turned into Queens when they reach the opposite side of the board.

The game nearly always ends in a check-mate.

The seven levels give a good wide variety of play, and are explained below.

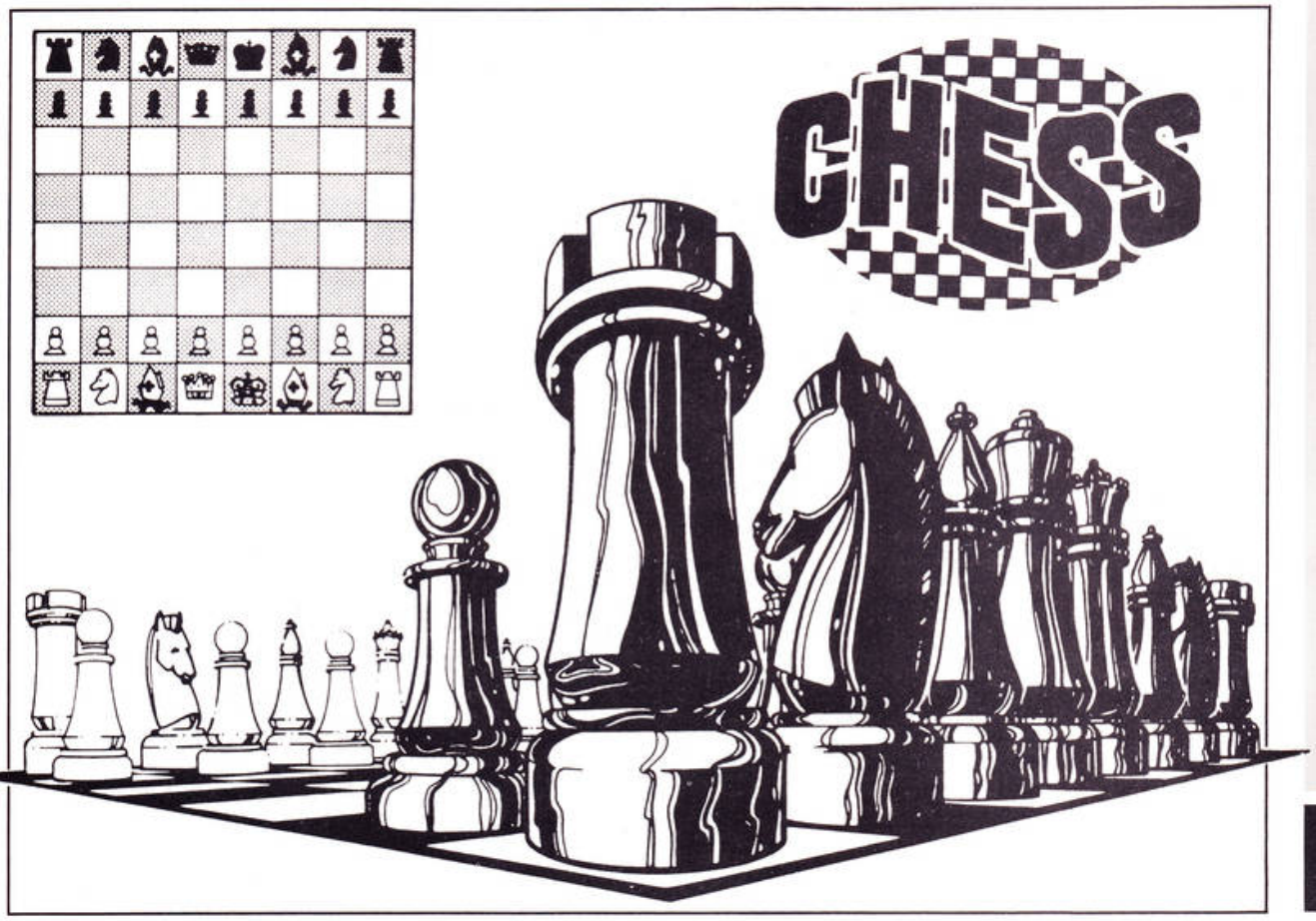
Level 0	replies in 2 seconds and is the easiest level.
Level 1	replies in 15 seconds and plays reasonably.
Level 2	replies in 40 seconds and is suitable for casual players.
Level 3	replies in 3 minutes and plays an above average game.
Level 4	replies in 5 minutes, playing a game difficult to beat.
Levels 5 and 6	take longer. play very strongly, and arem ost suitable for correspondance games.

The combined efforts of my family have not yet been able to defeat level 3 without having to have help from the computer, obtained by pressing the 'M' for 'move' (or as I like to call it, 'mother') key. I have a negative attitude to this facility, because when things get rough it tempts the innocent player into cheating. Some would argue that it helps to learn the game.

This program may be considered worth-while to the serious chess-player, but is not worth the extra expense for a beginner, or a player who will only use the program for the occasional game, as this program contains sophisticated programming, rather than user 'friendliness'.

ZX CHESS II costs £9.95 for hte 16K ZX81, and is available from Artic Computing Ltd., 396 James Reckitt Avenue, Hull, N. Humberside. HU8 0JA.

NICHOLAS WILDING

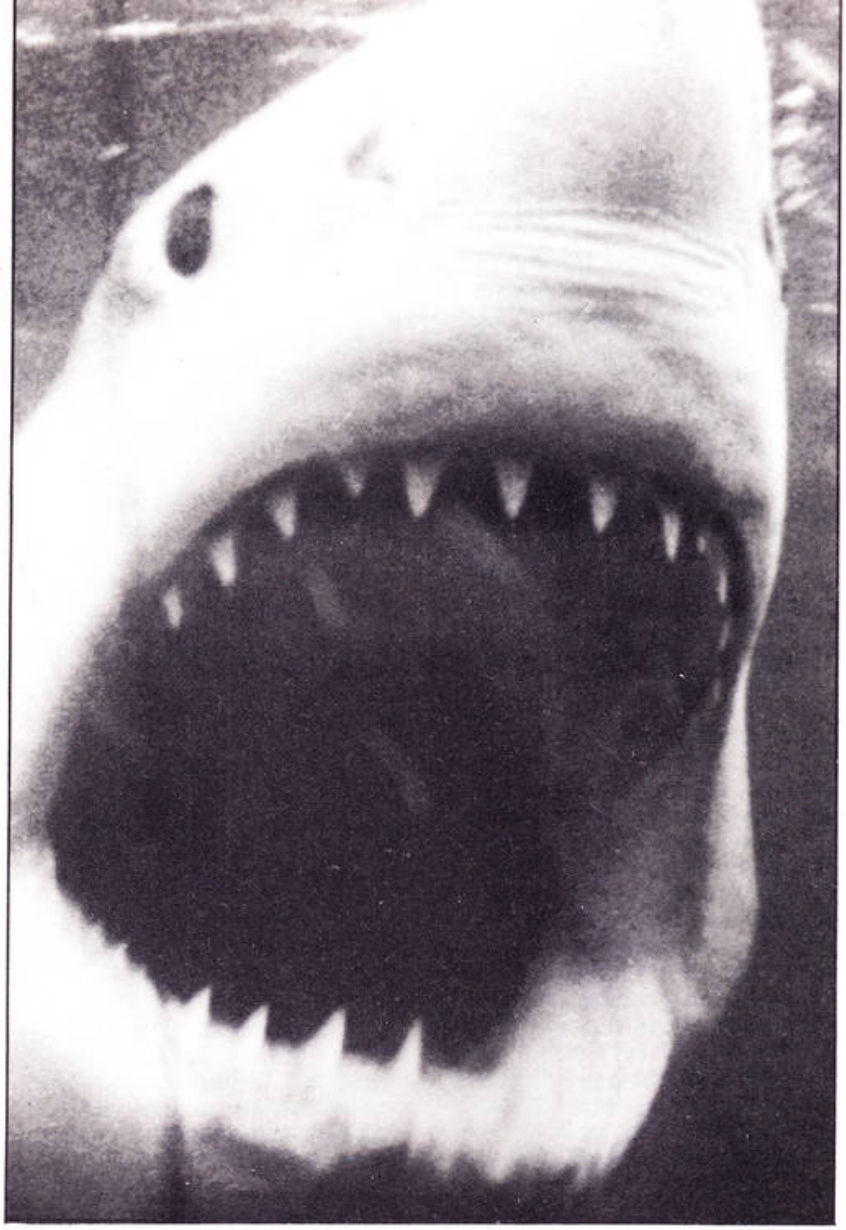


This great arcade game was written in Spectrum BASIC on the 16K Spectrum. After typing in the listing save it by typing SAVE"AQUASHARK" LINE 1 and 'ENTER'. Now just type 'RUN' and you will hear some melody. This means that the micro is defining the UDGs. Then three pages of explanation will appear and your beloved Spectrum will ask for your chosen skill level (level 9 is the easiest).

Keep swimming

Then comes the fun part — the game itself. You are a shark, who, being somewhat hungry, has to eat all the fish in the aquarium. However, there is another shark who is even more hungry and is out to try to eat you! (You didn't think there wouldn't be a catch, did you?) It's really as simple as that — if you have eaten all the fish, the melody starts playing and you are presented with a new screen. If you're beaten by that nasty other shark, you'll see another screen which will tell you how the game went (as if you didn't know) and it will ask if you want another game.

Be careful — that other fishy character can do very unexpected jumps! So if you're feeling sufficiently hungry, start typing in the game and start hunting.

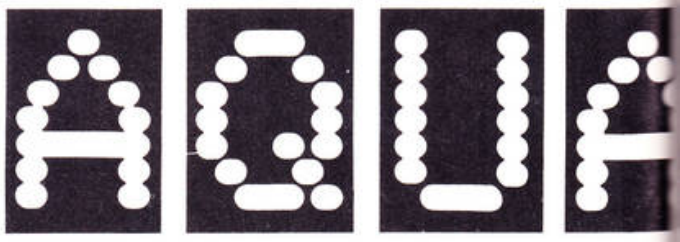


Rather than be chased by nasty beasties, just try your hand (or fin!) at being a shark to catch others.

Program structure

LINES:	FUNCTION:
1-4	set variables and explanation
5-92	set screen
95-106	main loop
200-220	own shark up
300-310	own shark down
400-410	own shark left
500-510	own shark right
600-690	logic-mover other shark
2000-2070	screen after game (melody too)
3000-3050	level-choice
4000-4020	melody after cleared screen
5000-5040	two pages explanation
8000-8050	set UDG

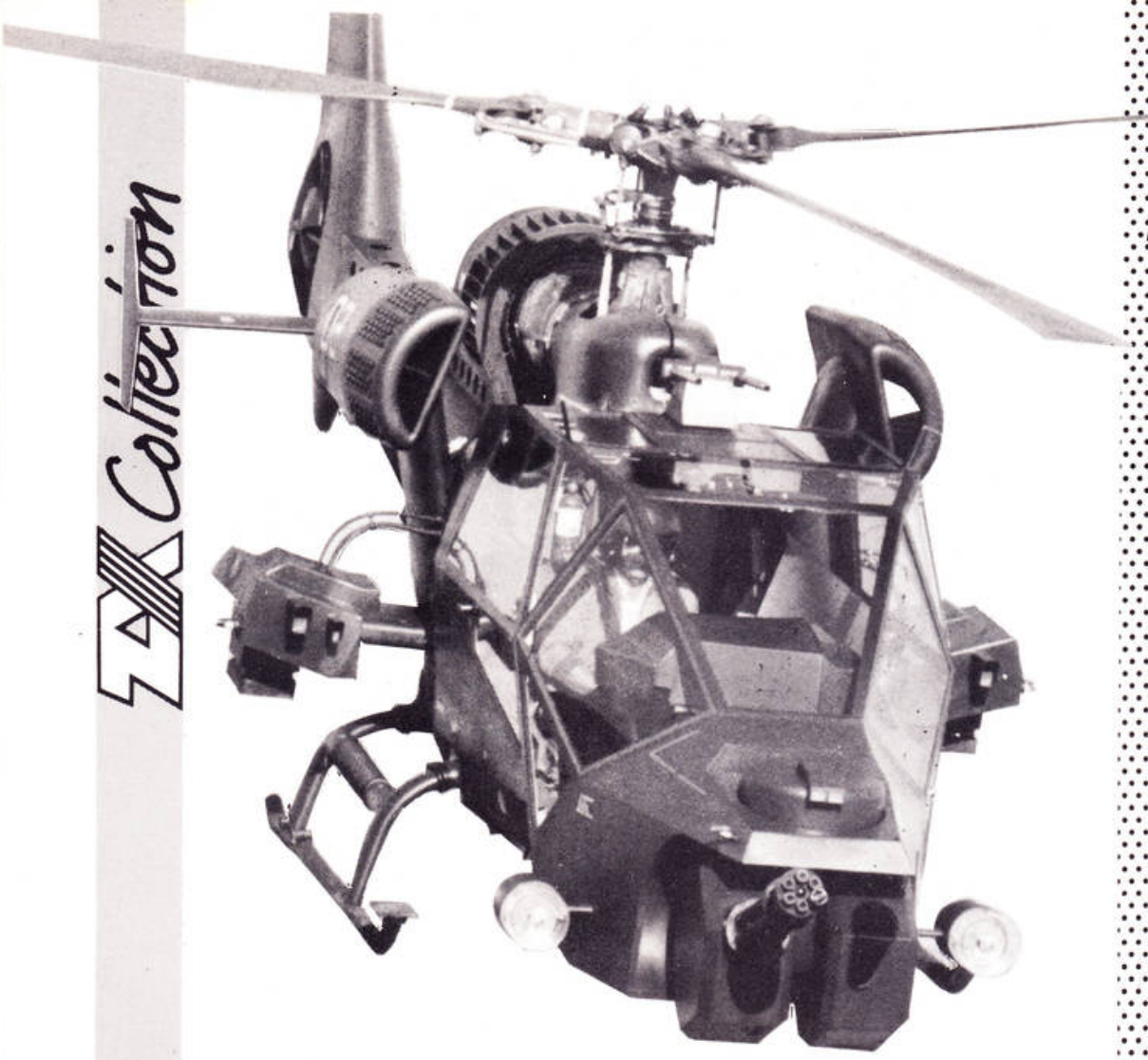
VARIABLES:	USE:
cnt	control subroutine
i	first loop-counter
j	second loop-counter
hsc	high score
sc	score
x, y	coordinates from own shark
x1, y1	coordinates from other shark
v, w, v1, w1	variables for branch-checking



```
  1 LET cnt=0: GO SUB 8000
  2 GO SUB 5000
  3 LET hsc=0: GO SUD 3000
  4 LET sc=0
  5 PAPER 4: BORDER 4: INK 0: C
LS : FLASH 0: BRIGHT 0
  9 PRINT AT 0,0;"**************
******************"
 10 PRINT AT 0,0; FLASH 1; INK
2;"score=";sc;AT
 0,14; FLASH 1; INK 2;"high scor
e=";hsc
 11 PRINT AT 2,20;"■"
 15 FOR j=0 TO 6 STEP 3
 20 FOR i=1+j TO 26 STEP 6
 30 PRINT AT 3+j*2,i;"┌ ┐"
 40 PAPER 5
 50 PRINT AT 4+j*2,i;"|  |"
 60 PRINT AT 5+j*2,i;"└──┘"
 70 PAPER 4
 75 IF cnt=1 THEN  LET cnt=0: R
ETURN
 80 NEXT i
 85 NEXT j
 90 LET cnt=1: LET j=6: LET i=1
: GO SUB 30
 92 PRINT AT 19,0;"┌";AT 19,31;
"┐";AT 20,0; PAPER 5;"|
                          |";AT 21,0;
"└──────────────────────────────
──┘"
 95 LET v=0: LET w=0: LET y=2:
LET x=20: LET x1=26: LET y1=16:
LET v1=1
 96 GO SUB 600
 97 FOR i=1 TO t*2
 98 IF x1=x AND y1=y THEN  GO T
O 2000
 100 IF INKEY$="" THEN  NEXT i:
GO TO 96
 101 IF INKEY$="7" AND w=0 THEN
 LET v=0: GO SUB 200
 102 IF INKEY$="6" AND v=0 THEN
 LET w=0: GO SUB 300
 104 IF INKEY$="5" AND x>2 AND y
=0 AND POINT ((x*8)-1,(21-y)*8+2
)=0 THEN  GO SUB 400
 105 IF INKEY$="8" AND x<29 AND
y=0 AND POINT (((x+2)*8)+1,(21-y
)*8+2)=0 THEN  GO SUB 500
 106 NEXT i: GO TO 100
 200 FOR a=3 TO 1 STEP -1: BEEP
.007,y+15: LET y=y-1: PRINT AT y
,x;"■";AT y+1,x;"  ";: IF SCREEN
$ (y-1,x)<>" " THEN  LET w=1: RE
TURN
 210 NEXT a
 220 RETURN
 300 BEEP .006,20-y: LET y=y+1:
PRINT AT y,x;"■";AT y-1,x;"  ";
 IF POINT (x*8+1,(20-y)*8)=1 OR
POINT ((x+1)*8+1,(20-y)*8)=1 THE
N  LET v=1: RETURN
 305 IF POINT ((x*8)+4,(20-y)*8+
2)=1 THEN  LET sc=sc+10: PRINT A
T 0,0; FLASH 1;"score=";sc: FLASH 0
: IF (sc/140)-INT (sc/140)=0
THEN  GO SUB 4000: GO TO 11
 310 GO TO 300
 400 FOR a=1 TO 3: LET x=x-1: BE
EP .009,x: PRINT AT y,x;"■";AT
y,x+2;" "
 410 NEXT a: RETURN
 500 FOR a=1 TO 3: LET x=x+1: BE
EP .009,x: PRINT AT y,x;"■";AT
y,x-1;" "
 510 NEXT a: RETURN
 600 BEEP .005,20: PRINT AT y1,x
1;"  "
 601 IF v1=1 THEN  PRINT AT y1,x
1; PAPER 5;"■"
 610 IF x<(x1-5) THEN  LET x1=x1
-6: LET v1=POINT ((x1*8)+4,(21-y
1)*8+2)
 620 IF x>(x1+5) AND (x1)<26 THE
N  LET x1=x1+6
 630 LET w1=-3
 640 IF x>x1 THEN  LET w1=3
 650 IF y<(y1-5) THEN  LET y1=y1
-6: LET x1=x1+w1
 655 IF x1>27 THEN  LET x1=x1-6
 660 IF y>(y1+5) THEN  LET y1=y1
+6: LET x1=x1+w1
 665 IF x1>27 THEN  LET x1=x1-6
 670 IF x1=26 AND y1=10 THEN  LE
T x1=x1-3
 680 LET y1=POINT ((x1*8)+4,(21-
y1)*8+2)
 690 PRINT AT y1,x1; PAPER 5; IN
K 2;"■": RETURN
2000 PAPER 4: BORDER 4: CLS : IN
K 1
2001 DATA .13,3,.17,6,.45,9,.45,
9,.45,9,.37,14,.35,9,.05,11,.25,
9,.25,7,.3,6
2002 RESTORE 2000
2003 FOR i=1 TO 11: READ x,y: BE
EP x,y: NEXT i
2010 PRINT AT 0,5; PAPER 7; BRIG
HT 1;"A Q U A S H A R K"
2020 PRINT AT 2,1;"Too bad, you
 where caught by","a ";  INK 2;"■
"; INK 1;"."
2030 PRINT AT 4,1;"You caught ";
sc/10;" fishes, so your","score
is "; PAPER 7; BRIGHT 1;sc
2040 IF sc>hsc THEN  PRINT  PAPE
R 6; BRIGHT 1; INK 2; FLASH 1;AT
 7,0;"Your score is the highest!
!"; LET hsc=sc
2050 PRINT AT 9,0;"The highest s
core is "; PAPER 7; BRIGHT 1;hsc
2060 PRINT AT 11,5; PAPER 7; INK
 2; INVERSE 1; FLASH 1;"ANOTHER
GAME(Y,N)?"; PAUSE 0
2065 IF INKEY$<>"y" THEN  GO TO 4
2070 STOP
3000 PAPER 2: BORDER 2: CLS : IN
K 5
3010 PRINT AT 0,5; INK 0; PAPER
7; BRIGHT 1;"A Q U A S H A R K"
3020 PRINT AT 5,0; INK 6;"_____
_____";AT 9
,0;"_____
_____"
3030 PRINT AT 7,5;"LEVEL (1-9)-9
=EASY"
3040 PAUSE 0: LET b$=INKEY$: IF
b$<"1" AND b$>"9" THEN  LET t=
CODE (b$)-48: RETURN
3050 GO TO 3040
4000 DATA .20,0,.20,0,.15,0,.15,
4,.20,7,.20,7,.25,4
4010 RESTORE 4000
4020 FOR i=1 TO 7: READ x,y: BEE
P x,y: NEXT i: RETURN
5000 PAPER 6: BORDER 7: CLS : IN
K 1
5010 PRINT  INK 2; BRIGHT 1;AT 3
,7;"A Q U A S H A R K"; BRIGHT 0
; INK 1;,,,"You are a ■ who hun
ts ■'s.  ",,,"Therefore you mus
t jump into the ",,,"┌ ┐",,,"  P
APER 5;"|  |"; PAPER 6;,,,"  PAPER
 5;"└──┘"; PAPER 6;
5020 PRINT ,,,"But beware, there
 is one ■ ","following you!!!",
,,,,;TAB 10; FLASH 1;"HIT ANY KE
Y"; PAUSE 0
5030 BORDER 6: CLS : PRINT AT 1,
10;"  7";TAB 10;"  |";TAB 10;"
  |";TAB 10;"  |";TAB 10;"  |
";TAB 8;"5----+----8";AT 7,10;"
  |";TAB 10;"  |";TAB 10;"  |"
;TAB 10;"  |";TAB 10;"  6";,,,
,,,;TAB 6; FLASH 1;"HIT ANY KEY
TO START"; PAUSE 0: CLS : BRIGHT
 0
5040 RETURN
8000 RESTORE 8000: FOR i=144 TO
155: FOR j=0 TO 7
8010 BEEP .03,-4+j+(143-i)*2: RE
AD z: POKE USR CHR$ i+j,z: NEXT
j: NEXT i
8011 BEEP .8,-29
8020 DATA 24,63,64,192,192,192,1
92,192,0,0,0,0,0,0,0,0,24,252,2,
3,3,3,3,3
8030 DATA 192,192,192,192,192,19
2,192,192,3,3,3,3,3,3,3,3,192,19
2,192,192,192,192,255,255,0,0,0,
0,0,0,255,255,3,3,3,3,3,3,255,25
5
8040 DATA 0,1,7,15,255,112,48,16
,126,249,255,222,192,0,0,0
8041 DATA 0,9,11,15,11,9,0,0,0,1
92,32,240,224,128,0,0
8050 RETURN
```

BY AYAL PINKUS

QUASHARK



UNLOADING THE CARGO

Unloading cargo may not seem the most exciting job in the world, but when a helicopter is just waiting to steal the crates you've unloaded, the situation becomes somewhat more interesting!

BY D G MENARY

A ship has just docked and you, the man with the trolley, must unload the cargo, one crate at a time, before the helicopter has a chance to steal any.

At the end of the pier there is a gun to fire at the helicopter. Sorry, but for every pass of the helicopter you only get three shots — any more and the game would be too easy.

Because the helicopter is bigger than you it can carry two crates of cargo to your one.

All instructions are explained in the program.

For people with issue one SPECTRUM you can change line 130 IF INKEY$="0" to IF IN 61436=254; the rest of the line is the same. Using the IN command enables you to keep your fingers on the key that controls the man with the trolley and fire the gun at the same time.

Breakdown of Program

Line 3-30	Set up variables.
Line 40-95	Set up screen. The only graphics used that are not user defined are graphics capshift 8 and in line 130 E (extended mode) symbol shift and keys Y and U.
Line 100-200	Main game loop.
Line 210-310	Scores and helicopter explosion, random helicopter appearance.
Line 400-440	End of game, high scores update and start game.
Line 500-750	User defined graphics.
Line 900-970	Instructions. While you read the first screen the user defined graphics are being set up, much better than a PLEASE WAIT on the screen.
Line 9990	To save the programme so that it will auto run, as a direct command GOTO 9990 ENTER.

```
  1 REM xxxxxxxxxxxxxxxxxxxxxxxx
        xUnderlined characterx
        xare entered in       x
        xGRAPHICS mode.       x
        xxxxxxxxxxxxxxxxxxxxxxxx
  2 PRINT AT 21,7; INK 3;"S T O
 P  T A P E ": PAUSE 200: GO TO
900
  3 LET j=0
  4 LET h=0
  5 BORDER 4: PAPER 1: INK 0: C
LS
  7 LET p=50
  8 LET x=0
  9 LET v=0
 10 LET a=0
 11 LET u=0
 12 LET e=0
 20 LET b=0
 30 LET c=25
 40 PRINT  PAPER 2; INK 6;AT 18
,0;"EEEEEEEEEEEEEEEEEEEEEEEEEE"
 50 PRINT  INK 6;AT 17,24;"Q"
 55 FOR r=19 TO 21: PRINT  PAPE
R 4;AT r,0;"
     ": NEXT r
 57 FOR u=0 TO 25 STEP 3: PRINT
  PAPER 4; INK 6;AT 19,u;"N": NE
XT u
 60 PRINT  PAPER 2; INK 6;AT 17
,0;"MPQ";AT 17,3;"MMPQQ";AT 16,3
;"MM QQ"; PAPER 4;AT 16,0;"S■R";
AT 15,3;"S■■■R"
 61 PRINT  PAPER 3; INK 0;AT 17
,0;"MMPQQ"; INK 2; PAPER 4;AT 16
,0;"S■■■R";AT 15,0;"      "
 62 PRINT  PAPER 1; INK 2;AT 17
,13;"MPQ";AT 16,13;"MPQ";AT 15,1
3;"MPQ"; PAPER 1;AT 14,13;"S■R"
 63 PRINT  PAPER 2; INK 7;AT 17
,16;"MQMPQMQ"; PAPER 1; INK 0;AT
 16,16;"S■■■■■R"
 64 PRINT  PAPER 1;AT 10,0;"SRS
RSR"; PAPER 3;AT 11,0;"MQMQMQ";A
T 12,0;"MQMQMQ";AT 13,0;"MQMQMQ"
; INK 4;AT 14,0;"■■■■■■■";AT 15,0
;"■■■"
 65 PRINT  PAPER 1; INK 5;AT 13
,6;"S■■■R"; PAPER 7; INK 0;AT 14
,6;"MMPQQ"
 70 PRINT  INK 7;AT 0,17;AT 1,1
5;"COPTERS SHOT";AT 1,0;"CARGO S
AVED"
 75 PRINT  PAPER 1; INK 7;AT 4,
15;"Cargo nicked"
 80 PRINT  INK 5; PAPER 0;AT 20
,26;"■■■■■■■";AT 21,26;"■■■■■■■";
INK 1;AT 19,26;"R    S"; PAPER 3
; INK 0;AT 18,27;"MQ";AT 17,28
;"MQ"
 85 PRINT  PAPER 6; INK 0;AT 21
,25;"E";AT 20,25;"E";AT 21,0;"EE
EEEEEEEEEEEEEEEEEEEEEEEE"
 87 PRINT  PAPER 5; INK 1;AT 20
,27;"CARGO"
 89 PRINT  PAPER 5; INK 1;AT 0,
0;"CARGO HIGH     COPTERS HIGH
   "
 90 PRINT  PAPER 5; INK 1;AT 0,
12;j
 95 PRINT  PAPER 5; INK 1;AT 0,
28;h
 99 GO TO 210
100 FOR f=2 TO 29 STEP 2
110 IF d<>1 THEN  GO TO 135
115 BEEP 0.01,-20
120 PRINT  INK 2;AT 8,f;"  BB"
125 IF b>2 THEN  GO TO 131
130 IF INKEY$="0" THEN  PLOT 19
2,40: DRAW  INK 7;-70,90: PLOT
OVER 1;192,40: DRAW  OVER 1;-70,
90: BEEP 0.015,20: LET b=b+1: IF
 f=14 THEN  GO TO 300
131 IF f<>22 THEN  GO TO 135
132 FOR g=9 TO 15 STEP 2: PRINT
  INK 2;AT g,c;"BB";AT g-2,c-1;"
  "
133 IF f=22 THEN  LET c=c+1: PR
INT AT 8,24;"    ": BEEP 0.009,60
: NEXT g: BEEP .01,10: BEEP .02,
-10: PRINT AT 15,28;"    ": FOR
g=14 TO 8 STEP -2: PRINT  INK 2;
AT g,27;"BB";AT g+2,27;"  ": NEX
T g: LET p=p-2: LET e=e+2
134 IF u=1 THEN  GO TO 150
135 PRINT  INK 1; PAPER 4;AT 20
,a;" CD "
140 IF u=0 THEN  GO TO 160
150 PRINT  INK 1; PAPER 4;AT 20
,a;" CD "
160 IF INKEY$="2" THEN  LET a=a
+1
165 IF INKEY$="1" THEN  LET a=a
-1
170 IF a>21 THEN  LET a=21
175 IF a<0 THEN  LET a=0
180 IF a=21 THEN  LET u=1
185 IF a=0 AND u=1 THEN  LET v=
v+1: PRINT  INK 7;AT 1,12;v: LET
 u=0: PRINT  INK 1; PAPER 4;AT 1
9,2;"E"
200 NEXT f
210 LET d=INT (RND*3)
212 PRINT  INK 2;AT 16,28;"E"
215 LET b=0
217 LET c=25
218 PRINT  PAPER 1;AT 8,30;"  "
```

```
 220 PRINT  PAPER 5; INK 1;AT 21
,28;p-v;"  ": IF p-v<1 THEN  GO
TO 400
 240 PRINT  INK 7;AT 1,28;s;"  "
 250 PRINT  PAPER 4;AT 19,2;" "
 260 PRINT  PAPER 4; INK 1;AT 20
,24;"E"
 270 PRINT  INK 7;AT 4,28;e
 280 GO TO 100
 300 PRINT  INK 6;AT 8,16;"##";A
T 8,16;"  ": FOR x=9 TO 14: PRIN
T  INK 6;AT 16-x,25-x;"I";AT 16-
x,25-x;" ";AT 16-x,8+x;"E";AT 16
-x,8+x;" ";AT x,25-x;"J";AT x,25
-x;" ";AT x,8+x;"L";AT x,8+x;" "
: BEEP RND*.02,x: NEXT x
 305 LET s=s+1
 310 GO TO 210
 400 PRINT #0;"   C A R G O  U N
 L O A D E D  "

T 4,1;v;AT 4,4;"crates of cargo
as against  the copters ";49-p
1;" crates of cargo"," I suggest
 you try again."
 422 IF v>j THEN  PRINT  PAPER 5
; INK 1;AT 0,12;v
 423 IF v>j THEN  LET j=v
 425 IF s>h THEN  PRINT  PAPER 5
; INK 1;AT 0,28;s: LET h=s
 428 PRINT  INK 7;AT 8,8;"Press
any key to unload";AT 9,8;"anoth
er ship of cargo."
 430 PAUSE 0
 440 CLS : GO TO 7
 500 RESTORE 550
 510 FOR i=USR "a" TO USR "u"+7
 520 READ q: POKE i,q
 530 NEXT i
 550 DATA 7,0,192,192,255,255,1,
0
 560 DATA 255,32,112,206,193,193
,255,252
 570 DATA 48,48,120,124,58,41,41
,49
 580 DATA 0,0,0,0,0,255,74,132
 590 DATA 170,0,85,0,170,0,85,0
 600 DATA 0,0,0,126,126,126,126,
126
 610 DATA 128,64,32,16,15,13,27,
31
 620 DATA 0,0,48,15,15,48,0,0
 630 DATA 1,2,15,255,0,0,0,0
 640 DATA 216,223,48,63,0,0,0,0
 650 DATA 224,16,28,255,0,0,0,0
 660 DATA 3,252,248,0,0,0,0,0
 670 DATA 128,159,146,146,158,12
8,128,128
 680 DATA 66,36,36,36,36,36,
66,2
 690 DATA 126,126,126,126,126,25
5,74,132
 700 DATA 0,60,36,36,36,36,36,36
 710 DATA 1,121,73,73,121,1,1,1
 720 DATA 128,192,224,240,248,25
2,254,255
 730 DATA 1,3,7,15,31,63,127,255
 740 DATA 192,56,4,194,34,17,17,
8
 750 DATA 4,24,33,70,136,144,16,
32
 800 PAUSE 0: GO TO 1000
 900 BORDER 1: PAPER 1: INK 5: C
LS
 910 PRINT "There's a copter aft
er the cargofrom the ship thats
just docked."
 915 PRINT
 920 PRINT "You need this cargo
to survive, but can only manage
one crate ata time."
 925 PRINT
 930 PRINT "The copter can manag
e two cratesso to help you, ther
e's a gun atthe end of the pier.
"
 935 PRINT
 940 PRINT "   Problem is there'
s only three shots  for  every
pass  of the copter. So don't be
 too trigger happy"
 945 PRINT
 950 PRINT "Fifty crates of carg
o to unload and quite a few copt
ers will tryto nick it all."
 960 PRINT  INK 7;AT 21,0;"Press
 any key to continue"
 970 GO TO 500
1000 CLS
1010 PRINT  INK 2;AT 2,4;"The ro
tten copter   AB"
1020 PRINT  INK 1; PAPER 4;AT 4,
4;"You and your trolly CD "
1030 PRINT  INK 6;AT 6,4;"The th
ree shot gun  G"
1040 PRINT  INK 5; PAPER 3;AT 8,
4;"Some cargo EEEEEEEEEEEE"
1050 PRINT AT 11,5;"C  O  N  T
R  O  L  S"
1060 PRINT AT 13,0;"key  1  to m
ove CD to the left"
1070 PRINT AT 15,0;"key  2  to m
ove CD to the right"
1080 PRINT AT 17,3;"key  0  to f
ire the gun  G"
1090 PRINT  INK 7;AT 21,0;"Press
 any key to start game"
1095 PAUSE 0: GO TO 3
9999 SAVE "cargo" LINE 1
```

RENUMBER

This machine code program will renumber the lines in your programs and alter any affected statements accordingly.

Here is a renumber program written in machine code for the Spectrum. It renumbers all lines and alters GOTO, GOSUB, RESTORE, LIST, LLIST, RUN and LINE statements accordingly. The program takes up just over 2/3K at the top of memory. It is as fast as any commercially available renumber program, although it does a bit extra, by renumbering all commands which can have line numbers as their parameters (ie not just GOTO and GOSUB commands).

We give you the listings for both the 16K and 48K versions. Each listing will place the Renumber machine code above RAMtop and below the user defined graphics for the Spectrum (the machine code can only go into these locations, it is not relocateable). Each listing incorporates a routine for testing its own DATA statements which can be used by entering GOTO 700. This routine takes 30 seconds to run and performs three tests on the data in order to determine if the data arecorrect (see 16K listing). There is also a routine for saving the machine code.

Get typing

Once you have typed in the program and are satisfied that the data are correct, you may run the program. The machine code is then ready to use and you will not require the BASIC any more (it is only needed initially, to place the machine code in memory). You may save the machine code by entering GOTO 500 (Microdrive) or GOTO 600 (tape). I would suggest that you also save the BASIC, just in case.

Note

Whenever you want to load the renumber code, you must enter CLEAR 64664 (48K) or CLEAR 31896 (16K) before loading, otherwise the Spectrum will crash.

Up and running

To run the renumber program, simply enter RANDOMIZE USR 64665 (48K) or RANDOMIZE USR 31897 (16K). If there is a program to renumber, the prompt: Enter renumbering step (MAX=number)

0000

will appear. Type in the number (if you make a mistake type it in again — you will notice that the number will rotate from right to left as you press each key). Press Enter when you have finished or "H" to exit. After a few seconds or milliseconds depending on the size of program (a full 40K program can take up to 15 seconds to renumber), the computer will be returned to BASIC. You will notice that your previously alphabet-soup-like line numbers are now neat and structured. GOTO, GOSUB, LIST, LLIST, RESTORE,RUN and LINE commands are also altered to suit these new line numbers, although if these commands have no parameter or have a variable as a parameter, they will be left alone. If any of these commands point to a line number beyond the end of the program (GOTO 9000 when the last line in the program is 7000), they will be changed to point to line 9991. This is the largest prime number available for line numbers and therefore unlikely to be occupied after renumbering; you can then put a STOP statement in line 9991 to trap such GOTO and GOSUB commands.

BY MATTHEW HOMER

```
 1
 2 REM  48K version
 3
 20 CLEAR 64664
 30 FOR a=64665 TO 65366
 40 READ b: POKE a,b: NEXT a
 50 DATA 17,0,0,213,42,83,92,35
,35,237,91,75,92,235,237,82
 60 DATA 235,202,190,252,250,19
0,252,78,35,70,9,17,3,0,25,209
 70 DATA 19,213,195,162,252,209
,237,83,138,253,17,5,0,237,82,23
7,66,86,35,94,237,83,140,253
 80 DATA 33,48,255,6,30,126,35,
215,16,251,62,255,33,0,0,237
 90 DATA 91,138,253,237,82,17,1
5,39,235,1,0,0,25,3,56,252
 100 DATA 237,82,11,120,177,200,
237,67,150,253,205,27,26,33,78,2
55
 110 DATA 6,9,126,215,35,16,251,
62,143,50,237,90,62,185,50,242
 120 DATA 90,33,146,253,62,48,6,
4,119,35,16,252,195,55,253,122
 130 DATA 184,202,78,253,120,33,
147,253,17,146,253,1,3,0,237,176
 140 DATA 198,48,18,214,48,87,62
,22,215,62,1,215,62,14,215,33
 150 DATA 146,253,6,4,126,215,35
,16,251,24,2,22,10,62,191,219
 160 DATA 254,203,103,200,203,71
,202,152,253,62,247,219,254,205,
127,253
 170 DATA 175,184,32,187,62,239,
219,254,205,127,253,175,184,40,2
20,62
 180 DATA 1,184,32,4,6,0,24,167,
62,11,144,71,24,161,6,5
 190 DATA 95,62,16,163,200,15,16
,251,201,0,0,0,0,0,0,0
 200 DATA 0,0,0,0,0,10,0,42,93,9
2,34,142,253,33,146,253
 210 DATA 205,254,25,42,142,253,
34,93,92,33,0,0,237,74,40,134
 220 DATA 237,91,150,253,19,237,
82,242,55,253,237,67,150,253,62,
15
 230 DATA 50,237,90,50,242,90,42
,83,92,35,35,34,142,253,35,237
 240 DATA 75,138,253,35,62,236,1
90,40,77,60,190,40,73,62,229,190
 250 DATA 40,68,62,225,190,40,63
,62,202,190,40,58,62,240,190,40
 260 DATA 53,62,247,190,40,48,62
,13,190,32,19,17,3,0,25,34
 270 DATA 142,253,35,11,151,184,
32,203,185,32,200,195,9,255,60,1
90
 280 DATA 32,193,17,6,0,25,24,18
8,33,7,39,229,195,118,254,193
 290 DATA 42,144,253,195,213,253
,197,35,34,144,253,62,57,150,250
,32
 300 DATA 254,198,245,56,234,1,6
,0,62,14,237,177,226,32,254,62
 310 DATA 5,145,35,35,94,35,86,2
29,42,140,253,237,82,250,25,254
 320 DATA 221,42,150,253,42,83,9
```

RENUMBER

12345

```
2,70,35,78,235,237,66,40,20,250
 330 DATA 116,254,9,235,35,78,35
,70,35,9,237,75,150,253,221,9
 340 DATA 195,88,254,221,229,209
,225,114,43,115,235,14,0,71,221,
33
 350 DATA 145,253,17,24,252,205,
241,254,17,156,255,205,241,254,1
7,246
 360 DATA 255,205,241,254,125,20
5,253,254,121,197,144,40,62,242,
192,254
 370 DATA 237,68,42,144,253,6,0,
79,197,205,232,25,193,6,0,42
 380 DATA 142,253,94,35,86,235,2
37,66,235,114,43,115,195,220,254
,79
 390 DATA 6,0,42,144,253,197,205
,85,22,193,6,0,42,142,253,94
 400 DATA 35,86,235,9,235,114,43
,115,195,220,254,193,6,0,237,91
 410 DATA 144,253,33,146,253,237
,176,193,235,17,5,0,25,195,212,2
53
 420 DATA 175,25,60,56,252,237,8
2,61,95,177,200,123,12,198,48,95
 430 DATA 121,50,7,255,221,115,0
,201,42,83,92,237,75,138,253,197
 440 DATA 237,91,150,253,151,193
,185,32,2,184,200,11,197,114,35,
115
 450 DATA 35,78,35,70,35,9,237,7
5,150,253,235,9,235,24,230,22
 460 DATA 0,0,69,110,116,101,114
,32,114,101,110,117,109,98,101,1
14
```

```
 470 DATA 105,110,103,32,115,116
,101,112,40,77,65,88,61,41,22,1,
13,62,22,1,18,60
 480 STOP
 500 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 502 REM ▚ TO SAVE ON MICRODRIVE ▚
 503 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 504 REM
 510 SAVE *"m";1;"renumber"CODE
64665,702
 520 VERIFY *"m";1;"renumber"COD
E
 530 STOP
 600 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 602 REM ▚▚  TO SAVE ON TAPE  ▚▚
 603 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 604 REM
 610 SAVE "renumber"CODE 64665,7
02
 620 STOP
 700 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 710 REM ▚▚▚ TEST ROUTINE. ▚▚▚
 718 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 719 REM
 720 LET c=0: LET d=1: LET e=1
 725 FOR a=64665 TO 65366
 730 READ b
 740 LET c=c+b: LET d=d*(b/255+.
6): LET e=e*(b/(a-64000)+.9)
 750 NEXT a
 755 BEEP .5,30
 760 IF c=82731 AND STR$ d="9.59
28184" AND STR$ e="66038.365" TH
EN  PRINT "YOUR DATA IS CORRECT.
": STOP
 770 PRINT "ERROR IN DATA."
```

RENUMBER

67890

```
 1
 2 REM  16K Version
 3
 20 CLEAR 31896
 30 FOR a=31897 TO 32598
 40 READ b: POKE a,b: NEXT a
 50 DATA 17,0,0,213,42,83,92,35
,35,237,91,75,92,235,237,82
 60 DATA 235,202,190,124,250,19
0,124,78,35,70,9,17,3,0,25,209
 70 DATA 19,213,195,162,124,209
,237,83,138,125,17,5,0,237,82,23
7
 80 DATA 66,86,35,94,237,83,140
,125,33,48,127,6,30,126,35,215
 90 DATA 16,251,62,255,33,0,0,2
37,91,138,125,237,82,17,15,39
 100 DATA 235,1,0,0,25,3,56,252,
237,82,11,120,177,200,237,67
 110 DATA 150,125,205,27,26,33,7
8,127,6,9,126,215,35,16,251,62
 120 DATA 143,50,237,90,62,185,5
0,242,90,33,146,125,62,48,6,4
 130 DATA 119,35,16,252,195,55,1
25,122,184,202,78,125,120,33,147
,125
 140 DATA 17,146,125,1,3,0,237,1
76,198,48,18,214,48,87,62,22
 150 DATA 215,62,1,215,62,14,215
,33,146,125,6,4,126,215,35,16
 160 DATA 251,24,2,22,10,62,191,
219,254,203,103,200,203,71,202,1
52
 170 DATA 125,62,247,219,254,205
,127,125,175,184,32,187,62,239,2
19,254
 180 DATA 205,127,125,175,184,40
,220,62,1,184,32,4,6,0,24,167
 190 DATA 62,11,144,71,24,161,6,
5,95,62,16,163,200,15,16,251
 200 DATA 201,0,0,0,0,0,0,0,0,0,
0,0,0,10,0,42
 210 DATA 93,92,34,142,125,33,14
6,125,205,254,25,42,142,125,34,9
3
 220 DATA 92,33,0,0,237,74,40,13
4,237,91,150,125,19,237,82,242
 230 DATA 55,125,237,67,150,125,
62,15,50,237,90,50,242,90,42,83
 240 DATA 92,35,35,34,142,125,35
,237,75,138,125,35,62,236,190,40
 250 DATA 77,60,190,40,73,62,229
,190,40,68,62,225,190,40,63,62
 260 DATA 202,190,40,58,62,240,1
90,40,53,62,247,190,40,48,62,13
 270 DATA 190,32,19,17,3,0,25,34
,142,125,35,11,151,184,32,203
 280 DATA 185,32,200,195,9,127,6
0,190,32,193,17,6,0,25,24,188
 290 DATA 33,7,39,229,195,118,12
6,193,42,144,125,195,213,125,197
,35
 300 DATA 34,144,125,62,57,150,2
50,32,126,198,245,56,234,1,6,0
 310 DATA 62,14,237,177,226,32,1
26,62,5,145,35,35,94,35,86,229
 320 DATA 42,140,125,237,82,250,
25,126,221,42,150,125,42,83,92,7
0
 330 DATA 35,78,235,237,66,40,20
,250,116,126,9,235,35,78,35,70
```

RENUMBER

12345

```
 340 DATA 35,9,237,75,150,125,22
1,9,195,88,126,221,229,209,225,1
14
 350 DATA 43,115,235,14,0,71,221
,33,145,125,17,24,252,205,241,12
6
 360 DATA 17,156,255,205,241,126
,17,246,255,205,241,126,125,205,
253,126
 370 DATA 121,197,144,40,62,242,
192,126,237,68,42,144,125,6,0,79
 380 DATA 197,205,232,25,193,6,0
,42,142,125,94,35,86,235,237,66
 390 DATA 235,114,43,115,195,220
,126,79,6,0,42,144,125,197,205,8
5
 400 DATA 22,193,6,0,42,142,125,
94,35,86,235,9,235,114,43,115
 410 DATA 195,220,126,193,6,0,23
7,91,144,125,33,146,125,237,176,
193
 420 DATA 235,17,5,0,25,195,212,
125,175,25,60,56,252,237,82,61
 430 DATA 95,177,200,123,12,198,
48,95,121,50,7,127,221,115,0,201
 440 DATA 42,83,92,237,75,138,12
5,197,237,91,150,125,151,193,185
,32
 450 DATA 2,184,200,11,197,114,3
5,115,35,78,35,70,35,9,237,75,15
0,125,235,9,235,24,230,22
 460 DATA 0,0,69,110,116,101,114
,32,114,101,110,117,109,98,101,1
14
 470 DATA 105,110,103,32,115,116
,101,112,40,77,65,88,61,41,22,1,
13,62,22,1,18,60
 480 STOP
```

```
 500 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 502 REM ▚TO SAVE ON MICRODRIVE▚
 503 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 504 REM
 510 SAVE *"m";1;"renum"CODE 318
97,702
 520 VERIFY *"m";1;"renum"CODE
 530 STOP
 600 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 602 REM ▚▚  TO SAVE ON TAPE  ▚▚
 603 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 604 REM
 610 SAVE "renum"CODE 31897,702
 620 STOP
 705 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 709 REM ▚                    ▚
 710 REM ▚    TEST ROUTINE.   ▚
 711 REM ▚ This routine checks ▚
 712 REM ▚ the sum and product ▚
 713 REM ▚ and ordered product ▚
 714 REM ▚ of the data to find ▚
 715 REM ▚ out if an error has ▚
 716 REM ▚ been made in typing ▚
 717 REM ▚                    ▚
 718 REM ▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
 719 REM
 720 LET c=0: LET d=1: LET e=1
 725 FOR a=31897 TO 32598
 730 READ b
 740 LET c=c+b: LET d=d*(b/255+.
64): LET e=e*(b/(a-31200)+.9)
 750 NEXT a
 755 BEEP .5,30
 760 IF c=75179 AND STR$ d="1422
2.227" AND STR$ e="5.5839508" TH
EN  PRINT "YOUR DATA IS CORRECT.
": STOP
 770 PRINT "ERROR IN DATA."
```

RENUMBER

6789

WORDSEARCHING

Now anyone can find the words hidden in those grids of letters. This program will cope with any puzzle which does not exceed 32 columns or 20 rows in size. After the program has been loaded and run, the user is asked to type in the number of rows and columns in the puzzle. Each row of the puzzle must then be typed in, and the computer prints it out in the middle of the screen. The user is then prompted to type in the word to be searched for. If it is found, the word will flash to indicate its position; if not found the word will be printed at the bottom of the screen, along with the note "NOT FOUND". This will help you to check that the word was typed in correctly.

BY K WARD

With this clever Spectrum program anyone will be able to find the hidden words in those Wordsearch puzzles that magazines often print.

Program breakdown

The program uses two arrays, A$ to hold the letters of the puzzle and A( ) to hold the flashing attribute for the corresponding letters of array A$. Initially, the attributes are set at 0 when the array is dimensioned, but when the searched-for word is found the attribute for each letter is preceded by Flash 0 or Flash 1. In this way the computer not only locates the word, if it is present, but also shows where it is. Use is made of the Spectrum's useful feature of being able to re-dimension an existing array (line 80), so resetting all attributes back to 0. This is much easier and quicker than having a further routine to change all the 1's back to 0's before searching for the next word.

Line function

Lines 20-130	Set up the puzzle and matching arrays.
Lines 140-180	Allows you to input the word to be searched for.
Lines 190-210	Locates the first letter of the word and either branches to the search routine or prints word not found.
Lines 250-620	Searches for the word in the 8 possible directions from the initial letter.
Lines 700-750	Changes the flashing attributes in array A( ).
Lines 760-820	Prints out the puzzle.

List of variables

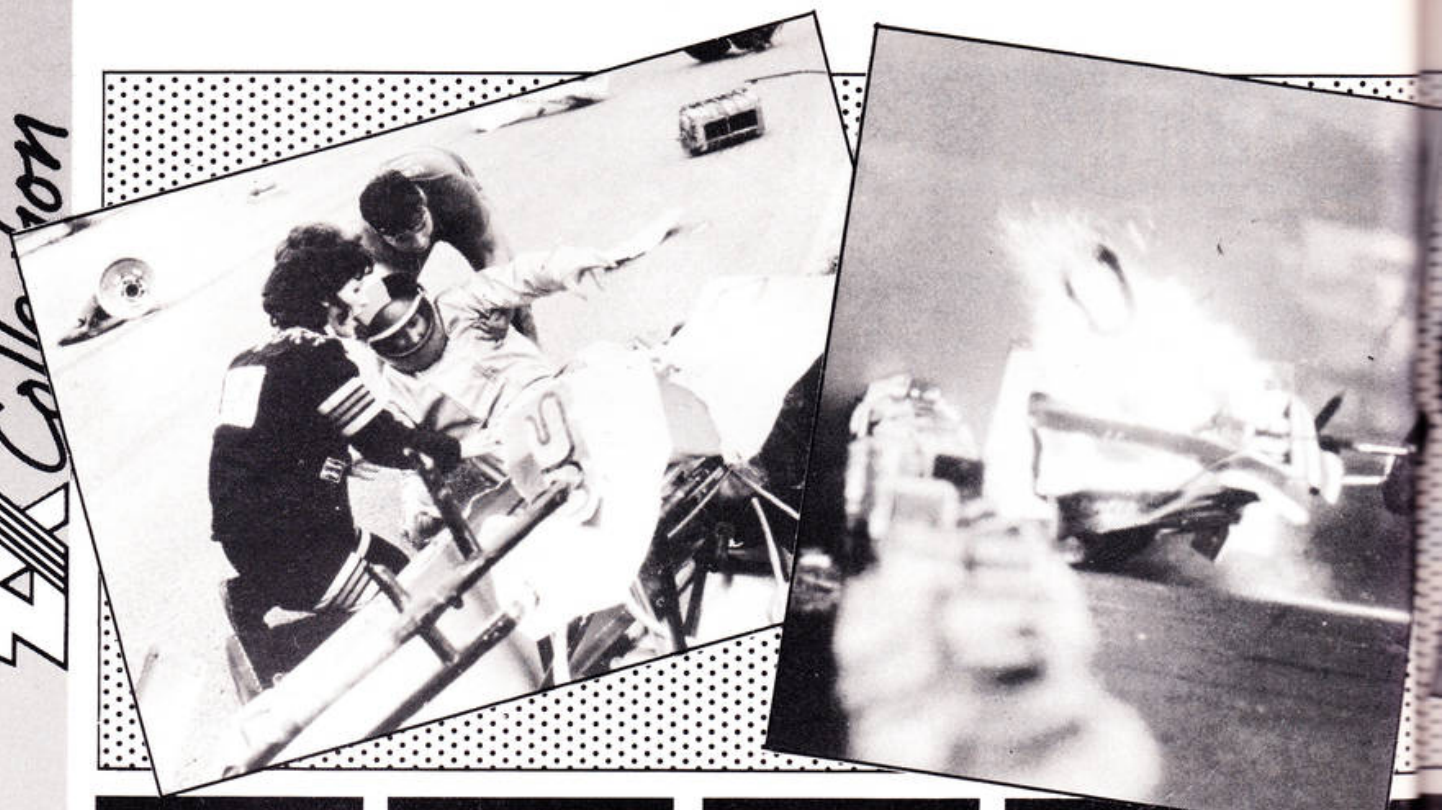
POKE 23658,8	Sets the Spectrum to Capital Mode.
COL	Number of columns in puzzle.
ROW	Number of rows in puzzle.
E$	Line of spaces for blanking out unwanted letters.
L$	Each row of puzzle.
X	Counter for rows.
Y	Counter for columns.
W$	Word to be searched for.
Z	Flashing attribute for printing.
R	
S	Loop counters.
C	
A$( )	Array to hold letters of puzzle.
A( )	Array to hold flashing attribute.

```
  10 REM ** WORDSEARCH **

  20 REM ** Initialize Table **
  30 POKE 23658,8
  35 BORDER 2: PAPER 2: INK 0
  40 CLS : PAPER 6: PRINT AT 0,5
;" ------------------ ";AT 1,5;
" W O R D S E A R C H ";AT 2,5;"
 ------------------ "
```

```
  50 PRINT '''"Please Enter:": PR
INT '"Number of COLUMNS (MAX 32)
 >> ";: INPUT COL: PRINT COL
  60 PRINT '"Number of ROWS    (
MAX 20) >> ";: INPUT ROW: PRINT
ROW
  65 IF COL>32 OR ROW>20 THEN  P
RINT AT 14,3; FLASH 1;" SQUARE T
OO BIG FOR SCREEN ": STOP
  70 DIM A$(ROW,COL)
```

```
  80 DIM A(ROW,COL)
  85 LET E$="
                "
  90 FOR R=1 TO ROW
 100 PRINT AT 16,6;"INPUT ROW ";
R;" PLEASE"
 110 INPUT L$: LET A$(R)=L$
 120 NEXT R
 130 PAPER 2: CLS : GO SUB 760

 140 REM ** Input Word **
 150 INPUT "WORD to find ";W$
 155 IF W$="" THEN  GO TO 150
 160 PRINT AT 20,0; PAPER 2;E$;E
$
 170 IF W$="0" THEN  STOP
 180 DIM A(ROW,COL)

 190 REM * Locate First Letter *
 200 LET X=1: LET Y=1
 210 IF A$(X,Y)=W$(1) THEN  GO T
O 250
 220 LET Y=Y+1: IF Y=COL+1 THEN
 LET Y=1: LET X=X+1
 230 IF X=ROW+1 THEN  PRINT AT 2
0,0;W$;AT 21,0; FLASH 1;" NOT FO
UND ": GO TO 140
 240 GO TO 210

 250 REM * Search Routine *
 260 IF Y>COL-LEN W$+1 THEN  GO
TO 300
 270 FOR S=1 TO LEN W$-1
 280 IF A$(X,Y+S)<>W$(1+S) THEN
 GO TO 300
 290 NEXT S: GO TO 700
 300 IF Y<LEN W$ THEN  GO TO 350
 310 FOR S=1 TO LEN W$-1
 320 IF A$(X,Y-S)<>W$(1+S) THEN
 GO TO 350
 330 NEXT S
 340 LET Y=Y-LEN W$+1: GO TO 700
 350 IF X>ROW-LEN W$+1 THEN  GO
TO 390
 360 FOR S=1 TO LEN W$-1
 370 IF A$(X+S,Y)<>W$(1+S) THEN
 GO TO 390
 380 NEXT S: GO TO 730
 390 IF X<LEN W$ THEN  GO TO 440
 400 FOR S=1 TO LEN W$-1
 410 IF A$(X-S,Y)<>W$(1+S) THEN
 GO TO 440
 420 NEXT S:
 430 LET X=X-LEN W$+1: GO TO 730
 440 IF X<LEN W$ OR Y>COL-LEN W$
+1 THEN  GO TO 480
 450 FOR S=1 TO LEN W$-1
 460 IF A$(X-S,Y+S)<>W$(1+S) THE
N  GO TO 480
 470 NEXT S: GO TO 740
 480 IF X>ROW-LEN W$+1 OR Y<LEN
W$ THEN  GO TO 530
 490 FOR S=1 TO LEN W$-1
 500 IF A$(X+S,Y-S)<>W$(1+S) THE
N  GO TO 530
 510 NEXT S
 520 LET X=X+LEN W$-1: LET Y=Y-L
EN W$+1: GO TO 740
 530 IF X>ROW-LEN W$+1 OR Y>COL-
LEN W$+1 THEN  GO TO 570
 540 FOR S=1 TO LEN W$-1
 550 IF A$(X+S,Y+S)<>W$(1+S) THE
N  GO TO 570
 560 NEXT S: GO TO 750
 570 IF X<LEN W$ OR Y<LEN W$ THE
N  GO TO 620
 580 FOR S=1 TO LEN W$-1
 590 IF A$(X-S,Y-S)<>W$(1+S) THE
N  GO TO 620
 600 NEXT S
 610 LET X=X-LEN W$+1: LET Y=Y-L
EN W$+1: GO TO 750
 620 GO TO 220

 700 REM ** Change Attributes **
 710 FOR C=Y TO Y+LEN W$-1
 720 LET A(X,C)=1: NEXT C: GO SU
B 760: GO TO 140
 730 FOR C=X TO X+LEN W$-1: LET
A(C,Y)=1: NEXT C: GO SUB 760: GO
 TO 140
 740 FOR C=0 TO LEN W$-1: LET A(
X-C,Y+C)=1: NEXT C: GO SUB 760:
GO TO 140
 750 FOR C=0 TO LEN W$-1: LET A(
X+C,Y+C)=1: NEXT C: GO SUB 760:
GO TO 140

 760 REM ** Printout **
 765 PAPER 6
 770 FOR R=0 TO ROW-1
 780 FOR C=1 TO COL
 790 LET Z=A(R+1,C)
 800 PRINT AT R+INT ((20-ROW)/2)
,C+INT ((30-COL)/2); INVERSE Z;A
$(R+1,C)
 810 NEXT C: NEXT R
 820 RETURN
```

WORDSEARCHING



CRASHER

BY JAMES DRURY

How do you fancy yourself as a top class rally driver? Well here's your chance to find out just how good your reflexes are when it comes to driving.

Drive your car around the snake-like track — without smashing into the bricks which are scattered about it. When typing in the listing bear in mind that the character in line 80 is Graphic A, and that in line 90 is Graphic BC. Save this, keep the tape at the finishing place, and type in the character saver. RUN, and you will get the usual prompt for SAVE. Make sure that the tape is wound to the position just after the main program and 'start the tape and press any key'. This will save the UDG's, which will auto-load if the main program is saved using: SAVE 'crasher' line 9999.

```
  10 FOR n=USR "a" TO USR "c"+7
  20 READ a: POKE n,a
  30 NEXT n
  40 DATA 0,62,8,62,28,62,28,0
  50 DATA 255,170,213,170,213,17
0,213,255
  60 DATA 255,171,85,171,85,171,
85,255
  70 SAVE "characters"CODE 65368
,167
```

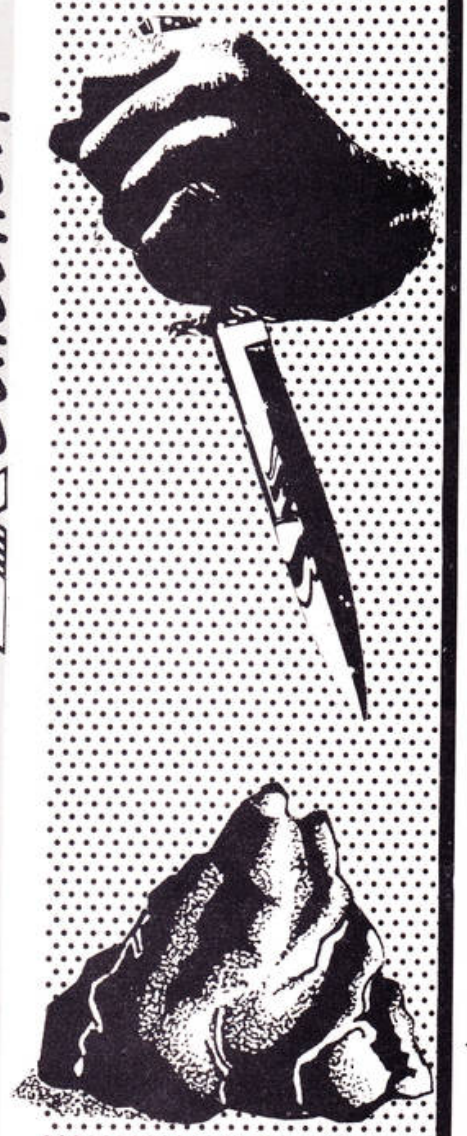
Line	contents
30-40	Set up variables and attributes
50-160	Main loop
115	Makes the screen 'auto-scroll'
155	Check to see whether the player is cheating
240-243	Check for 'RAST K'ey pressed by means of system for variable LAST K
500-550	Hi-score routine
9040-9080	Sets attributes
9999	Auto-run

NOTE: On the 16K Spectrum the number 65368 should be replaced by 32600 or USR 'a'.

Variables

c	car position
r	road position
sc	score
h	hi-score
rnd	not a function. This is where the new road is printed in relation to the old one.
1$(704)	If you print a screenful of spaces over the screen, you will get your PAPER and INK and FLASH and INVERSE and BRIGHT ATTRIBUTES shown up without the screen clearing.
s	the pitch of the BEEP while the game is being played


```
    1 REM **********************
          *Underlined characters*
          *are entered in       *
          *GRAPHICS mode.       *
          **********************
   10 DIM i$(704): LET h=100: GO
TO 230
   20 GO SUB 9500
   30 LET sc=0: LET r=10: LET c=1
2
   40 POKE 23692,12
   50 FOR s=0 TO 10
   60 LET rnd=INT (RND*2)
   70 LET r=r-(rnd=1 AND r>1)+(rn
d=0 AND r<20)
   80 PRINT AT 16,c; BRIGHT (sc>4
); PAPER 7; INK 2;"A"
   90 PRINT AT 21,r; BRIGHT 1; IN
K 2; PAPER 4;"BC"; PAPER 7;TAB r
+6; PAPER 4;"BC"
  100 PRINT
  110 LET c=c+(INKEY$="8")-(INKEY
$="5"): IF PEEK 23560=57 THEN  G
O TO 110
  115 POKE 23560,54
  120 IF SCREEN$ (16,c)<>" " THEN
   GO TO 200
  130 LET sc=sc+1
  140 BEEP .005,s
  150 NEXT s
  155 IF ATTR (16,c)<>122 THEN  G
O TO 200
  160 GO TO 40
  200 PRINT AT 16,c; FLASH 1; INK
 0; PAPER 4;"A"
  205 PRINT AT 10,10; INK 6; PAPE
R 2; FLASH 1;"You scored ";sc;AT
 12,9;"High score:";h
  210 FOR n=10 TO 40 STEP .4: BEE
P .001,n: NEXT n
  220 IF sc>h THEN  GO SUB 500
  230 GO SUB 9500: PRINT AT 13,0;
 FLASH 1; INK 5; PAPER 1;"     P
ress '5' to play or '8' to
                              stop/
re-start noise.              ";AT 1
,10;"High score:";h
  235 PRINT AT 5,0; INK 2; PAPER
5;"      AAAAAAAA
         A      A
         A      A
         A      A
         AAAAAAAA
         A      A
         A      A
         A      A
         A      A
   "
  240 POKE 23560,0: FOR n=3 TO 6
STEP .5: BEEP .007,n: OUT 254,IN
T (RND*7): NEXT n
  241 IF PEEK 23560=56 THEN  PAUS
E 0: GO TO 245
  242 IF PEEK 23560=53 THEN  GO T
O 20
  243 GO TO 240
  245 POKE 23560,0
  247 IF PEEK 23560=53 THEN  GO T
O 20
  248 IF PEEK 23560=56 THEN  GO T
O 240
  249 GO TO 247
  500 GO SUB 9500: PRINT AT 10,1;
  FLASH 1; PAPER 2; INK 4;"
    A high score!          "
  530 PRINT AT 12,5; PAPER 5; INK
  3; FLASH 1;"The high score was:
";h;AT 14,5;"The high score is :
";sc;AT 16,10;"Press a key"
  540 RESTORE 9040: FOR n=1 TO 15
: READ z: READ x: BEEP x,z: IF I
NKEY$="" THEN  NEXT n: GO TO 510
  550 LET h=sc: RETURN
9040 DATA 0,.3,0,.2,5,1.25
9050 DATA 0,.3,5,.18,9,1.25
9060 DATA 0,.3,5,.18,9,.4
9070 DATA 0,.3,5,.18,9,.4
9080 DATA 0,.3,5,.18,9,1.5
9500 FLASH 0: BRIGHT 1: PAPER 7:
 INK 1: BORDER 0: PRINT AT 0,0;
OVER 1;i$: BRIGHT 0: BORDER 0: C
LS : POKE 23624,0
9510 RETURN
9999 LOAD ""CODE USR "a": RUN
```



KNIFE

Three simple objects that can each overcome the other — we hope that you will be overcome with this fascinating children's game for the Spectrum.

BY COLIN GOOCH

This is a children's game, but can have a strange fascination for adults too — the challenge to 'outguess' the Spectrum. The game is very simple. Three objects are involved; a stone, which will blunt a knife, a knife which will wrap paper, and a sheet of paper which can wrap a stone. Thus each object can overcome one of the others. The Spectrum chooses an object, you choose an object and they are both displayed on the screen. A point is given to the victor and we repeat the process. A game consists of ten goes. A check is kept of these rounds as well. In a draw situation no-one scores.

the provision to simply load the next game. . . this allows anyone to load something different without screaming for you to load it. I do this with all the tapes of my programs that members of the family are likely to use.

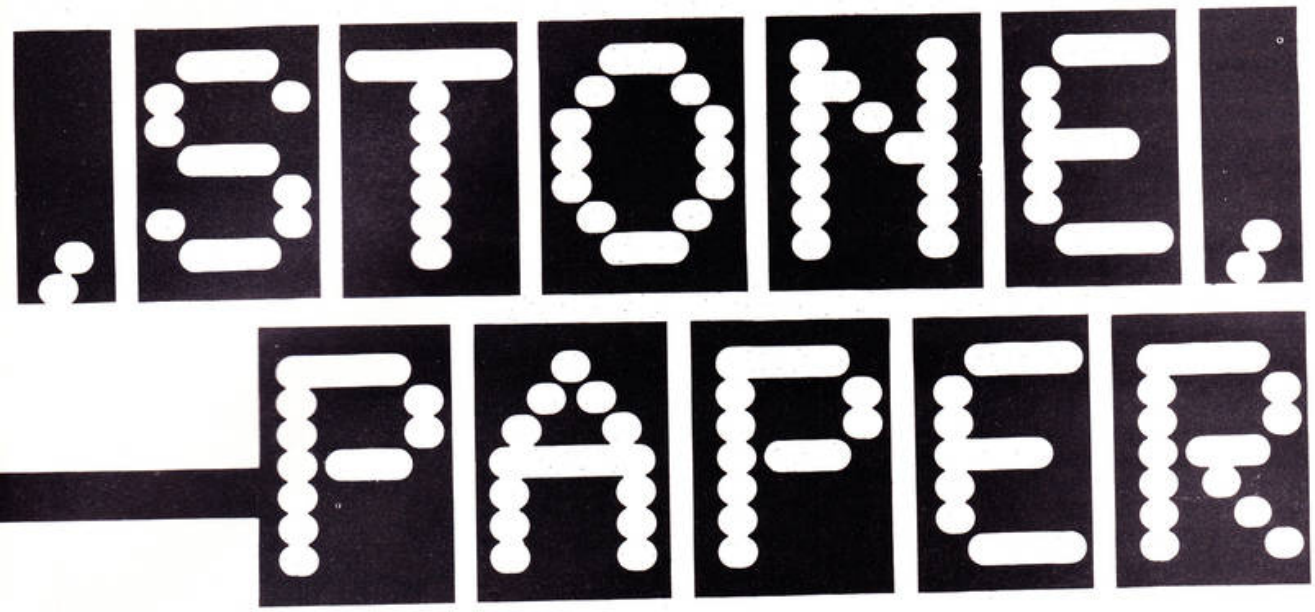
Variables

A$ B$	Blank arrays to clear parts of screen.
C$	Guesses
K$ Z$	Temporary variables
L$	Spectrum Guesses
M$	Player's Guesses
N$ O$	Captions
A	Subroutine to be jumped to.
M,N	Position of display on screen.
PL	Player
pX	
PS	Player's score
sX	
SS	Spectrum's score
PRO	Player round score
SRO	Spectrum round score

Program function

Lines 100-1060	Deal with initialisation. Printing out the instructions and setting up the screen display, by calling the required subroutines.
Lines 1070-1260	Are the main control loop. Your choice of object is made at line 1070 and the Spectrum's at line 1120 under normal circumstances. This can be overidden by the lines between these two points. This happens if a player tries to 'beat the system' by constantly entering the same answer on the basis that by choosing randomly the Spectrum can only be right one out of three attempts. This is the same sort of action that a human opponent would take. The rest of the loop assesses the scores, prints them out, then clears the screen display ready for the next round.
Lines 1270-1280	Subroutine to ask for your choice.
Lines 1290-1450	Print out the graphics.
Lines 1460-1510	Screen display.
Lines 1520-1600	Deal with the end game, printing out the scores and offering another go.
Lines 1610-1720	Instructions including the defining of User Defined Graphics.

As can be seen the main part of the program is the assessing of scores and producing the screen display, the actual decision making being limited to half a dozen lines, but particularly in a game for children it is well worth going for a attractive display, and a routine that will not crash with a wrong input. Also at the end is




```
1010 BORDER 3: PAPER 1: CLS : PR
INT AT 10,8; PAPER 5; FLASH 1;"S
TOP THE TAPE";AT 14,4; FLASH 0;"
PRESS ENTER TO CONTINUE": INPUT
 LINE Z$
1020 BORDER 1: PAPER 4: INK 0: C
LS : RANDOMIZE : POKE 23658,8
1030 DIM A$(96): DIM B$(608): LE
T N$="I WIN": LET O$="YOU WIN":
LET C$=" KNIFE  STONE  PAPER "
1040 CLS : GO SUB 1610
1050 DIM M(2): LET M(1)=1: LET M
(2)=17: LET N=6: LET PL=1: LET S
S=0: LET PS=0: LET SRO=0: LET PR
O=0
1060 GO SUB 1460
1070 GO SUB 1270: LET M$=L$: LET
 PL=1: BEEP .1,30
1080 LET C$=C$(1 TO (LEN C$-20) )
1090 IF C$=" STONE  STONE  STONE
 " THEN  GO SUB 1370: GO TO 1130
1100 IF C$=" KNIFE  KNIFE  KNIFE
 " THEN  GO SUB 1320: GO TO 1130
1110 IF C$=" PAPER  PAPER  PAPER
 " THEN  GO SUB 1400: GO TO 1130
1120 LET A=1+INT (RND*3): GO SUB
 1320+50*(A=2)+90*(A=3)
1130 LET C$=C$+M$
1140 IF L$=" STONE " AND M$=" KN
IFE " THEN  PRINT AT 16,7; PAPER
 5;"STONE BLUNTS KNIFE ";AT 18,4
; PAPER 7; FLASH 1;N$: LET SS=SS
+1
1150 IF M$=" STONE " AND L$=" KN
IFE " THEN  PRINT AT 16,7; PAPER
 5;"STONE BLUNTS KNIFE ";AT 18,2
0; PAPER 7; FLASH 1;O$: LET PS=P
S+1
1160 IF L$=" PAPER " AND M$=" ST
ONE " THEN  PRINT AT 16,7; PAPER
 5;"PAPER WRAPS STONE ";AT 18,4
; PAPER 7; FLASH 1;N$: LET SS=SS+
1
1170 IF M$=" PAPER " AND L$=" ST
ONE " THEN  PRINT AT 16,7; PAPER
 5;"PAPER WRAPS STONE ";AT 18,20
; PAPER 7; FLASH 1;O$: LET PS=PS
+1
1180 IF L$=" KNIFE " AND M$=" PA
PER " THEN  PRINT AT 16,7; PAPER
 5;" KNIFE CUTS PAPER ";AT 18,4;
 PAPER 7; FLASH 1;N$: LET SS=SS+
1
1190 IF M$=" KNIFE " AND L$=" PA
PER " THEN  PRINT AT 16,7; PAPER
 5;" KNIFE CUTS PAPER ";AT 18,20
; PAPER 7; FLASH 1;O$: LET PS=PS
+1
1200 IF L$=M$ THEN  PRINT AT 16,
13; PAPER 7; BRIGHT 1; FLASH 1;"
 DRAW "
1210 PRINT AT 20,4; PAPER 7; INK
 1;" SCORE=";SS;AT 20,20;" SCORE
=";PS
1220 IF INKEY$<>"" THEN  GO TO 1
470
1230 PRINT AT 21,6;"
       "; INPUT "PRESS ENTER FOR
 ANOTHER GO"; LINE Z$: BEEP .1,1
0
1240 IF SS=10 OR PS=10 THEN  GO
SUB 1530
1250 FOR V=4 TO 20: PRINT AT V,1
;"
": NEXT V
1260 GO SUB 1490: GO TO 1070
1270 PRINT AT 21,6; PAPER 7; INV
ERSE 1;" PRESS S or K or P "
1280 LET K$=INKEY$: IF K$="S" TH
EN  LET PL=2: GO SUB 1320: RETUR
N
1290 IF K$="P" THEN  LET PL=2: G
O SUB 1380: RETURN
1300 IF K$="K" THEN  LET PL=2: G
O SUB 1400: RETURN
1310 GO TO 1280
1320 REM PRINT STONE
1330 PRINT AT N,M(PL)+3;"B■■■■■■
1340 PRINT AT N+1,M(PL)+3;"■■■■■
■■"
1350 PRINT AT N+2,M(PL)+3;"B■■■■
■■"
1360 LET L$=" STONE ": RETURN
1370 REM PRINT PAPER
1380 FOR B=-2 TO 4: PRINT AT N+B
,M(PL)+3; PAPER 7;"      ": NEXT
B
1390 LET L$=" PAPER ": RETURN
1400 REM PRINT KNIFE
1410 PRINT AT N+0,M(PL);"B■■■■■■
■■"
1420 PRINT AT N+1,M(PL);"■■■■■■■
■■"
1430 PRINT AT N,M(PL)+8; INK 5;"
■■■■■■"
1440 PRINT AT N+1,M(PL)+8; INK 5
;"■■■■■■"
1450 LET L$=" KNIFE ": RETURN
1460 REM PRINT SCREEN
1470 CLS : PRINT AT 0,0; PAPER 5
;A$
1480 PRINT AT 1,4; PAPER 7;" STO
NE    KNIFE    PAPER "
1490 PLOT 0,175: DRAW 255,0: DRA
W 0,-24: DRAW -255,0: DRAW 0,24:
 PLOT 1,174: DRAW 253,0: DRAW 0,
-22: DRAW -253,0: DRAW 0,22: PLO
T 128,152: DRAW 0,-152
1500 PRINT AT 14,3; PAPER 7; INK
 2;" SPECTRUM ";AT 14,20;" PLAYE
R "
1510 RETURN
1520 REM END ROUND
1530 CLS : GO SUB 1480: PRINT AT
 16,3+15*(PS=10); PAPER 7; FLASH
 1;" WINS ROUND ": FOR B=10 TO 3
0-60*(SS=10) STEP 1-2*(SS=10): B
EEP .1,B: NEXT B
1540 LET SRO=SRO+(SS=10): LET PR
O=PRO+(PS=10): PRINT AT 18,10; P
APER 7; INK 1;SRO;" ROUNDS ";PRO
1550 LET SS=0: LET PS=0
1560 INPUT " ENTER C TO CONTINUE
","  OR N FOR NEW PLAYER "," OR
 S TO STOP GAME "; LINE Z$: IF Z$
="C" THEN  GO TO 1060
1570 IF Z$="N" THEN  LET SRO=0:
LET PRO=0: GO TO 1060
1580 IF Z$="S" THEN  CLS : PRINT
 AT 10,2;" THANKS FOR PLAYING "
1590 PRINT AT 14,2;" TO LOAD NEX
T GAME START TAPE ": LOAD ""
1600 GO TO 1560
1610 REM INSTRUCTIONS
1620 GO SUB 1460: PRINT AT 3,0;
PAPER 6;B$
1630 PRINT AT 4,1; PAPER 6;" YOU
 CHOOSE ""STONE""  ""KNIFE""    "
;AT 5,1;"OR ""PAPER"" BY PRESSING
 ""S""  ""K""";AT 6,1;" OR ""P"""
1640 PRINT AT 10,1; PAPER 6;"""S
TONE"" BEATS ""KNIFE""";AT 11,6;
" BUT IS BEATEN BY ""PAPER"""
1650 PRINT AT 12,1; PAPER 6;"""K
NIFE"" BEATS ""PAPER""";AT 13,6;
" BUT IS BEATEN BY ""STONE"""
1660 PRINT AT 16,2; PAPER 7; INK
 3;" TRY TO  GUESS  WHAT THE ";A
T 17,2;" SPECTRUM WILL PLAY NEXT
 "
1670 INPUT "PRESS ENTER TO CONTI
NUE"; LINE Z$
1680 REM GRAPHICS
1690 DATA 3,15,63,63,127,127,255
,255,192,240,252,252,254,254,255
,255,255,255,127,127,63,63,15,3,
255,255,254,254,252,252,240,192
1700 RESTORE 1680: FOR N=0 TO 3:
 FOR M=0 TO 7
1710 READ A: POKE (USR CHR$ (65+
N))+M,A: NEXT M: NEXT N: RETURN
1720 STOP
1730 SAVE "STONE" LINE 1000
1740 PRINT AT 10,2;"SAVING COMPL
ETED ";AT 12,2;" REWIND TAPE. SE
T RECORDER TO ";AT 13,2;" PLAYBA
CK. START TAPE TO ";AT 14,12;" V
ERIFY ": VERIFY "STONE"
1750 CLS : PRINT AT 10,2;" PROGR
AMME VERIFIED "
```

A machine code program is used to provide a useful indexing system which can be searched for information at high speed.

BY K. ROBBINS

When setting out to write this program, the main requirement in mind was to produce a useful indexing system which could be searched for information at high speed. This requirement dictated a machine code program and since this was to be my first attempt at machine code programming, I decided for simplicity not to include any sorting routine prior to display. Such a routine could, in any case, be added to the BASIC part of the program and should not extend the time to access the data significantly because it would normally operate only on the few entries retrieved by the search.

Secondary requirements were that it should be easy to enter and to edit data. Since the data would be stored in consecutive memory addresses, I visualised that the latter could cause problems unless an extra "insert or delete data" were incorporated. At this point it was realised that the Sinclair ROM does this automatically when BASIC program lines are entered and furthermore that the ubiquitous REM statement will accept any of the alpha-numerical characters in the Sinclair list. Accordingly, each data entry is stored as a separate REM statement and can be extended, deleted, or moved elsewhere in the list very easily, by using the BASIC EDIT command. The penalty for this facility is that each entry requires six bytes over and above those comprising the entry itself. There are two for line number, two for number of characters in the text, one for REM and one for the closing NEWLINE. With normal length entries, this is not very significant and, using a 16K memory extension, approximately 250 entries with 50 data characters, including spaces, per entry can be stored.

In use

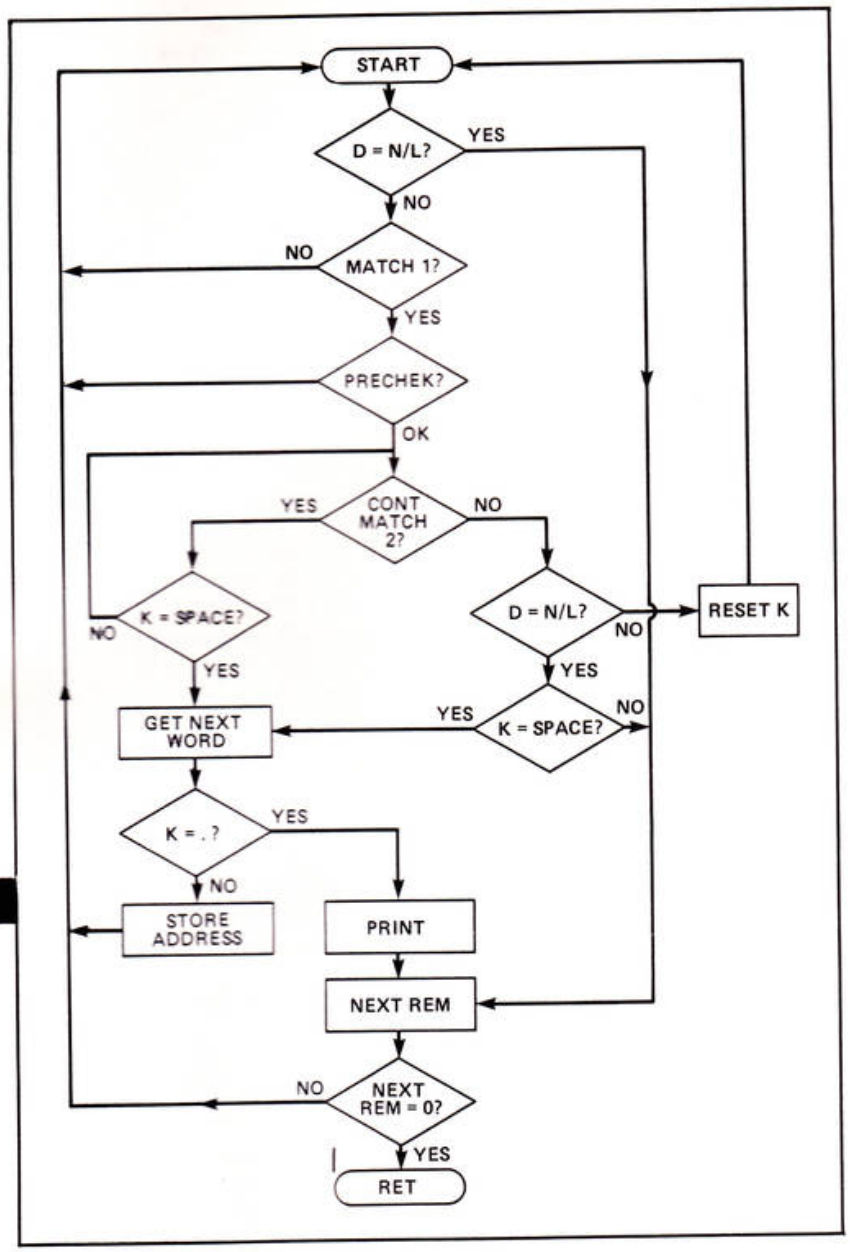
In use, the program is "fairly friendly", requesting whether you wish to add data, search the list, or list the entries. Points worth nothing here are that

(a) each entry may contain as many characters as desired
(b) the data list may be searched for one or more keywords. If more than one keyword is specified then only those entries containing all the keywords are displayed. The keyword list is limited to 41 characters, allowing on average approximately seven words. The keywords may be entered in any order.
(c) if the "list" option is selected, only the first "page" will be displayed. Continued listing requires the use of the BASIC LIST command in the usual way.

Machine code instructions range from the extremely simple to the very complex, depending on how many registers are affected and what happens to the flags. Fortunately, those used here are relatively simple and anyone should be able to follow them through if any debugging is required.

Logically speaking

The logic of the program caused me some trouble and to anyone beginning to write machine code programs, I cannot emphasise too strongly that you should be absolutely certain that you know what the next instruction should do and that the instruction that you write will do it! Right at the outset some sort of flow diagram should be drawn up, even if it is very simple — it can be extended and it serves to fix your ideas. This is particularly important with regard to the execution of loops and especially when these must be repeated with changes in initial data. Fig.1 shows a somewhat simplified diagram for this program, in which D refers to a data character and K to a keyword character. At the beginning of the routine, START and MATCH 1 get the first keyword and data characters and match them after first testing that the data character is not a NEWLINE. If the characters match then PRECHEK tests whether the preceding character was a REM or SPACE, if so the program continues (CONT). If not then the data character address is incremented repeatedly until a match does occur. On reaching CONT. both addresses are incremented and the characters tested for a match. If the characters match then the procedure is repeated until both are spaces, or the match falls. In this case, it may be because the data character is a NEWLINE. If so, the keyword character is



tested for a space and if found, the next keyword is obtained. If the characters have simply failed to match then the keyword character is reset to the start of the word, the data address is incremented and the routine goes back to START. As each new keyword is searched, a test is made to see whether it is a full stop. If so, the PRINT routine is entered and if not, the address is stored and the program returns to START. After passing through the PRINT routine. The program always determines the NEXREM address i.e., the address of the first character following REM in the next program line. This character is tested for SPACE and if found, the program presents the option list again, if not it returns to START and the whole sequence is repeated.

The actual program

The program itself is contained in a REM statement and the initial loading program is shown in Listing 1. This program should be entered in FAST mode. Listing 2 gives the addresses, mnemonics and the decimal machine codes. The latter, in the right hand column, are entered one at a time, followed by NEWLINE, using Listing 1. It will be found that there are ten spare bytes at the end of the machine code. These are the results of shortening the program during development and could have been removed, but it was not thought worth while to do so.

When the machine code has been entered,

POKE 16510,0

INDEX MC INDEX

```
   0 REM S$INKEY$ GOSUB ??RND, R
ETURN
A320 C  </ POKE ,  RETURN  C  RET
URN REM C  << NEW  <<??,  4  RETUR
N  C=/ NEXT RETURN
8199  C  /8E"INKEY$(,/SIN 7"INKEY
$/ E?RND7/ RUN Y
8999 REM
9000 PRINT "TO ADD DATA PRESS A.
            TO SEARCH FOR KEYWOR
DS PRESS K. TO LIST ENTRIES PRES
S L.        TO STOP PRESS BREAK.
"
9001 PRINT
9002 PRINT "(PRESS GOTO 1 AFTER
INPUT OF DATA)"
9003 POKE 16507,62
9004 POKE 16508,65
9005 POKE 16651,13
9008 POKE 16652,65
9010 IF INKEY$="" THEN GOTO 9010
9015 IF INKEY$="A" THEN GOTO 920
0
9017 IF INKEY$="L" THEN GOTO 924
```

```
9020 IF INKEY$="K" THEN PRINT "E
NTER KEYWORDS WITH ONE SPACE BET
WEEN EACH."
9025 INPUT K$
9030 LET K$=K$+" ."
9032 CLS
9035 FOR I=1 TO LEN K$
9040 POKE (16652+I),CODE K$(I)
9045 NEXT I
9055 RAND USR 16514
9060 GOTO 1
9200 LET N=N+1
9205 PRINT
9210 PRINT "NO. OF ENTRIES IS ";
N-2
9215 PRINT
9220 PRINT "TYPE ";N-1;" REM NEW
 ENTRY"
9230 STOP
9240 CLS
9250 LIST 1
9255 STOP
9260 SAVE "INDEX FRAME"
9300 GOTO 1
9350 REM INDEX FRAME
9500 REM C  K.ROLLINS 1983
```

as a direct command. This prevents the line containing the machine code from being edited or accidently modified. At this point it is advisable to save the program.

Next, delete lines 0 to 80 and type in the BASIC program given in listing 3, then type LET N=1, NEWLINE and GOTO 1.

In the listings, "sp" means one space. Where there appears to be a space in PRINT statements and one is not indicated, then a single key has been used to enter the word. This is done to save a little space and is achieved by typing THEN followed by the appropriate key and then deleting THEN.

As listed the program is self starting, but if option A, to add data is selected, then after entering the information followed as usual by NEWLINE, GOTO 1 must be entered to regain the menu.

In conclusion I would like to give some idea of the capabilities of the program. My first use for it was as an index for a collection of old vertical cut gramophone records, of which I have about 150. Each entry contained details of both sides of one record, including song title, artist, backing and catalogue number. This information usually occupies four or five lines of the display per entry. Experimentally, the last word of the last record was entered as a keyword. There was an almost imperceptible pause and then the last entry and one or two more containing the same keyword appeared on the screen. A previously written BASIC program took two minutes to the do the same job. When more than just a keyword is used, the pause before display extends to perhaps one or two seconds, caused presumably by the slowness of the BASIC in lines 9035-9045.

The program is self starting and after entering new data may be SAVEd by typing GOTO 9260. Now I wonder what I could use those ten spare bytes for?...

Listing 1

```
1     REM 182 zeroes
10    LET X = 16514
20    INPUT N
30    SCROLL
40    POKE X,N
50    PRINT X;"spsp";N
60    IF X = 16640
      THEN STOP
70    LET X = X + 1
80    GOTO 20
```



```
16514  START     LD HL 16653       33 13 65
16517            LD DE (16507)     237 9 123 64
16521            LD A (DE)         26
16522            CP NEWLINE        254 118
16524            JRZ NEXREM        40 80
16526            LD C (HL)         78
16527  MATCH 1   CP C              185
16528            JRZ PRECHEK       40 3
16530            INC DE            19
16531            JR 16521          24 244
16533  PRECHEK   DEC DE            27
16534            LD A (DE)         26
16535            CP O              254 0
16537            JRZ CONT          40 8
16539            CP REM            254 234
16541            JRZ CONT          40 4
16543            INC DE            19
16544            INC DE            19
16545            JR 16521          24 230
16547  CONT      INC DE            19
16548            INC DE            19
16549            INC HL            35
16550            LD C (HL)         78
16551            LD A (DE)         26
16552  MATCH 2   CP C              185
16553            JRNZ 16561        32 6
16555            CP O              254 0
16557            JRZ NEXT K        40 20
16559            JR 16548          24 243
16561            CP NEWLINE        254 118
16563            JRNZ RESET K      32 7
16565            LD A C            121
16566            CP O              254 0
16568            JRZ NEXT K        40 9
16570            JR NEXREM         24 34
16572  RESET K   LD HL (16651)     42 11 65
16575  NEXT D    INC DE            19
16576            LD A (DE)         26
16577            JR 16522          24 199
16579  NEXT K    INC HL            35
16580            LD A (HL)         126
16581            CP FULSTOP        254 27
16583            JRZ PRINT         40 5
16585            LD (16651) HL     34 11 65
16588            JR 16517          24 183
16590  PRINT     LD HL (16507)     42 123 64
16583            LD A (HL)         126
16594            CP NEWLINE        254 118
16596            JRZ 16602         40 4
16598            PRINT A           215
16599            INC HL            35
16600            JR 16593          24 247
16602            LD A NEWLINE      62 118
16604            PRINT A           215
16605            PRINT A           215
16606  NEXREM    LD HL (16507)     42 123 64
16609            LD B H            68
16610            LD C L            77
16611            DEC BC            11
16612            DEC BC            11
16613            DEC BC            11
16614            LD A (BC)         10
16615            LD E A            95
16616            INC BC            3
16617            LD A (BC)         10
16618            LD D A            87
16619            ADD HL DE         25
16620            INC HL            35
16621            INC HL            35
16622            INC HL            35
16623            INC HL            35
16624            LD (16507) HL     34 123 64
16627            LA A (HL)         126
16628            SUB A O           214 0
16630            JRZ RET           40 8
16632            LD HL 16653       33 13 65
16635            LD (16651) HL     34 11 65
16638            JR 16588          24 204
16640            RET               201
```

DAM ATTACK

BY NEIL WOOD

Playing games can be educational, as you'll find out with this program for the Spectrum.

Dam attack is an educational game for the Sinclair ZX Spectrum designed to teach simple addition and subtraction. It can be programmed to increase the child's speed of calculation and/or level of difficulty and is ideal for users of five to eleven years.

The game is fun and sometimes addictive but a supervisor is necessary to watch small children who may get angry with themselves if they keep getting the answer wrong, and for the older users in order to set the level of play at the beginning, which is an important part of the teaching abilities of the program. To my knowledge the idea for the game is original.

users. I wrote the program for my younger sister who adores it and it was my friends who told me more people could benefit from it.

2. The program listing itself shows good use of structure and programming.
3. To show that education is not boring anymore.
4. To give children the chance to play such programs at home and not just at school.

Instructions

Supervisor

Once the program has loaded a warning will sound to show the program has loaded and a page will appear titled: **Control.**

Three of the parameters which alter the play are self explanatory and include:

Speed: (1 fast to 40 slow):
Level: (1 easy to 30 hard):
Sums: (+ or −):

Two more are new to educational programs and are more of a customising section. They allow the sums to speed up and/or the level to get harder as the child plays. It works by increasing the speed or level after the child has answered so many sums correctly. It will appear on the screen as:

Level increase: 0 every 0 hits.
Speed increase: 0 every 0 hits.

If you do not want to use this section enter 0 for the appropriate values.

Entering the last number will take you to the main section where the child can take over.

Pressing EDIT will return you to this page.

Pupil

The program will display all writing in neat, easy to read, large characters and inputs are very simple.

On going into 'pupil mode' the screen will change to the play area as shown before and the words:

Press any key to start.

will appear. Do as it says and the game will start.

To answer the sums simply enter the numbers for the answer using keys 0 to 9 one at a time pressing 'p' if you make a mistake.

Your laser will not fire and destroy the sum until the correct answer has been found.

If you destroy the sum you will get another one but if you miss it will hit the dam and make a hole. Once the dam has been hit ten times the lower display will flood and the child will be offered another go.

A score is kept so that the child is forced to compete with other users.

The Game

The game is real time but very easily understood by small children. To the right of the display is a dam with water stored behind it. The clouds above are to make the program more life-like.

Sums will appear as missiles such as A/W and will fly at a chosen speed towards the dam. What the child has to do is simply answer the sum before it hits the dam and takes a small chunk out of it. After the dam has been hit ten times in a random order then the area in front of the dam will flood and the child gets the chance to play again at the same level.

The box in the bottom middle of the page or screen is where the child's answer will go in double height numbers so that it can be seen easily. Instructions for play are given later.

Aims

1. The main aim of course is to increase the addition and subtraction abilities of the child

Typing in the listing

The program consists of two parts so the following entry plan is very important.

First enter program A. This section allows the whole character set to be changed. Save this program using:

Save "GRAPHICS" Line 1.

Once the saving has finished keep the cassette on record for thirty more seconds. This is so that the program (A) has time to execute before program (B) is fed in.

Enter program B. This part is the actual game and routines. Save this using:

Save "PROGRAM" Line 1.

The game is now fully entered and can be played once loaded.

```
  10 REM
  20 REM
  30 REM       GRAPHICS
  40 REM
  50 REM
  55 REM    N-B-WOOD 1984
  56 REM
  60 LET a=PEEK 23675+256*PEEK 2
3676
  65 LET a=a-801
  80 CLEAR a
  85 PRINT AT 10,0; FLASH 1;"
   DO NOT STOP THE TAPE        "
  86 PRINT AT 0,0;
  90 LET a=PEEK 23730+256*PEEK 2
3731
  95 LET a=a+31
 100 RESTORE 110
 102 LET b=a/256
 104 FOR c=a-30 TO a-19: READ d:
 POKE c,d: NEXT c
 110 DATA 17,(b-INT b)*256,INT b
,33,0,61,1,0,3,237,176,201
 130 POKE 23606,(b-INT b)*256
 135 POKE 23607,INT b-1
 136 LET f=USR (a-30)
 145 FOR e=1 TO 43
 146 BEEP 0.1,e
 150 READ a$
 155 LET d=(CODE a$-32)*8+a
 160 FOR b=0 TO 7
 170 READ c
 180 POKE d+b,c
 190 NEXT b
 195 NEXT e
 200 FOR b=0 TO 159
 210 READ a
 220 POKE USR "a"+b,a
 230 NEXT b
 235 POKE 23609,25
 240 LOAD ""
 300 STOP
 500 DATA "!",0,56,126,127,255,2
55,255,255
 510 DATA "@",0,0,0,48,126,255,2
55,255
 520 DATA "#",0,0,0,0,120,255,25
5,255
 530 DATA "$",0,0,0,0,0,56,254,2
55
 540 DATA "%",255,127,127,63,0,0
,0,0
 550 DATA "&",255,255,255,255,0,
0,0,0
 560 DATA "'",255,255,254,252,0,
0,0,0
 570 DATA "(",1,3,7,15,31,63,127
,255
 580 DATA ")",128,192,224,240,24
8,252,254,255
```

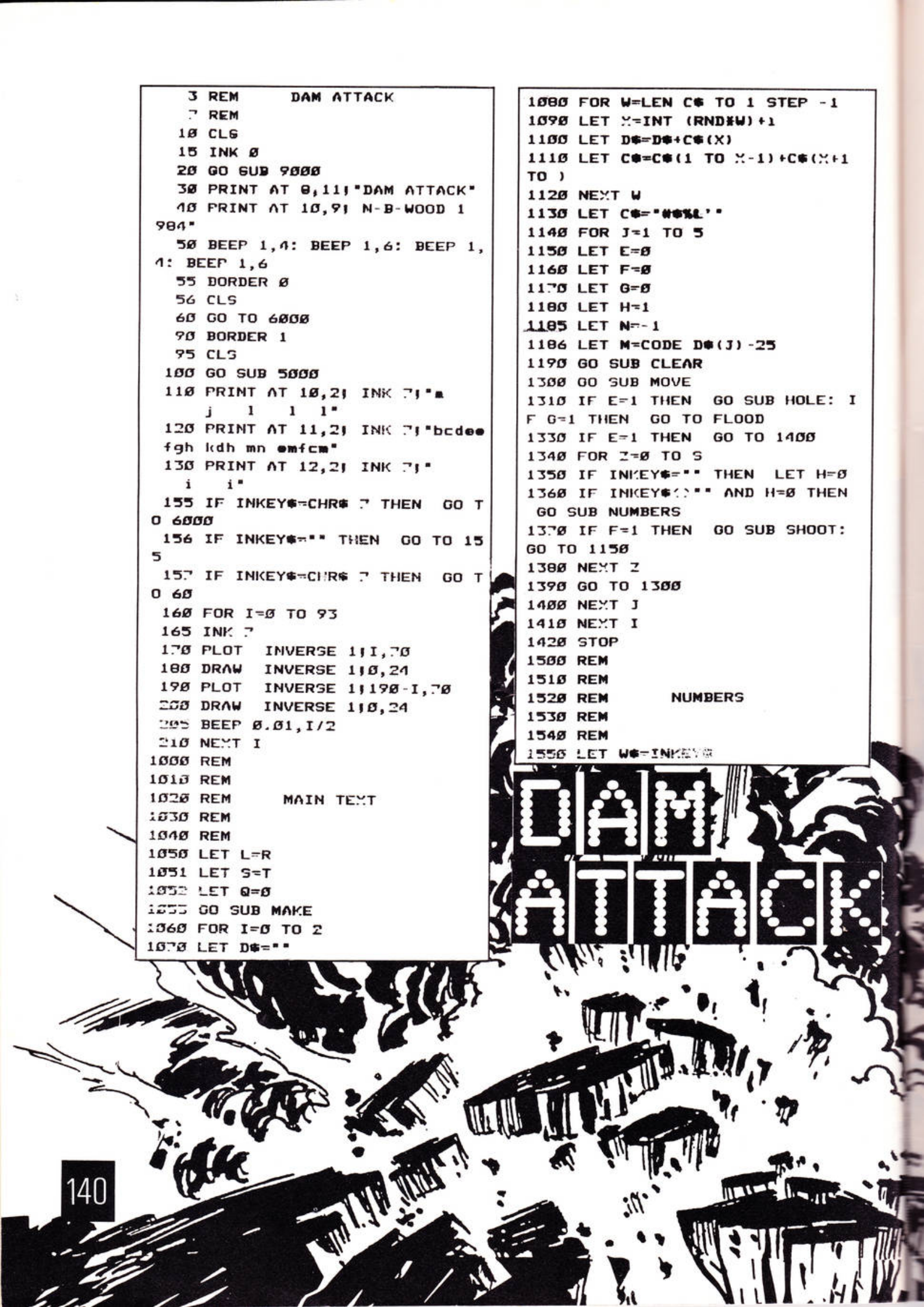
```
 590 DATA " ",24,24,24,24,24,60,
126,255
 600 DATA "\",2,199,235,122,202,
95,255,84
 610 DATA "[",195,227,115,127,12
7,115,227,195
 620 DATA "]",128,224,248,255,25
5,248,224,128
 630 DATA "a",0,0,126,65,65,65,6
5,65
 640 DATA "b",126,64,64,64,64,64
,64,64
 650 DATA "c",92,99,64,64,64,64,
64,64
 660 DATA "d",62,65,65,126,64,65
,65,62
 670 DATA "e",62,65,64,62,1,1,65
,62
 680 DATA "f",61,67,65,65,65,65,
67,61
 690 DATA "g",94,97,65,65,65,65,
65,65
 700 DATA "h",65,65,65,65,65,65,
65,63
 710 DATA "i",1,1,1,1,65,62,0,0
 720 DATA "j",0,0,64,64,68,72,72
,80
 730 DATA "k",80,96,96,80,80,72,
68,66
 740 DATA "l",0,0,32,32,32,32,32
,32
 750 DATA "m",124,32,32,32,32,32
```

```
,34,28
 760 DATA "n",62,65,65,65,65,65,
65,62
 770 DATA "o",0,0,60,34,65,65,65
,65
 780 DATA "p",65,65,65,65,65,65,
34,60
 790 DATA "q",65,65,65,65,65,65,
97,94
 800 DATA "r",0,0,0,0,0,16,16,0
 810 DATA "s",16,16,16,16,16,16,
16,8
 820 DATA "t",32,32,32,32,32,32,
17,14
 830 DATA "u",130,130,130,130,13
0,130,68,56
 840 DATA "v",126,65,65,65,65,65
,65,126
 850 DATA "w",64,64,64,64,64,64,
0,0
 860 DATA "x",0,0,32,32,32,32,32
,32
 870 DATA "y",32,32,32,32,32,32,
```

```
34,28
 880 DATA "z",62,65,65,65,65,65,
65,63
 890 DATA "#",1,1,1,1,65,62,0,0
 892 DATA ",",0,0,62,65,64,64,64
,64
 894 DATA ".",64,64,64,64,64,65,
65.62
 895 DATA "0",0,60,66,66,66,66,6
0,0
 910 DATA 0,4,12,28,60,108,12,12
 920 DATA 12,12,12,12,12,63,63,0
 930 DATA 0,30,63,99,99,6,6,12
 940 DATA 12,24,24,48,48,127,127
,0
 950 DATA 0,60,126,99,3,3,6,60
 960 DATA 60,6,3,3,99,126,60,0
 970 DATA 0,6,14,14,30,54,54,102
 980 DATA 102,127,127,6,6,6,6,0
 990 DATA 0,127,127,96,96,96,124
,126
1000 DATA 54,3,3,3,99,126,60,0
1010 DATA 0,3,6,12,24,24,48,48
1020 DATA 124,118,99,99,99,54,28
,0
1030 DATA 0,127,127,3,3,6,6,12
1040 DATA 12,24,24,48,48,96,96,0
1050 DATA 0,28,62,99,99,99,62,28
1060 DATA 54,99,99,99,99,54,28,0
1070 DATA 0,28,54,99,99,99,55,31
1080 DATA 6,6,12,12,24,48,96,0
1090 DATA 0,28,54,99,99,99,99,99
1100 DATA 99,99,99,99,99,54,28,0
```

```
    3 REM      DAM ATTACK
    7 REM
   10 CLS
   15 INK 0
   20 GO SUB 9000
   30 PRINT AT 0,11;"DAM ATTACK"
   40 PRINT AT 10,9; N-B-WOOD 1
984"
   50 BEEP 1,4: BEEP 1,6: BEEP 1,
4: BEEP 1,6
   55 BORDER 0
   56 CLS
   60 GO TO 6000
   90 BORDER 1
   95 CLS
  100 GO SUB 5000
  110 PRINT AT 10,2; INK 7;"a
      j    l    l   l"
  120 PRINT AT 11,2; INK 7;"bcdee
  fgh kdh mn emfcm"
  130 PRINT AT 12,2; INK 7;"
    i    i"
  155 IF INKEY$=CHR$ 7 THEN  GO T
O 6000
  156 IF INKEY$="" THEN  GO TO 15
5
  157 IF INKEY$=CHR$ 7 THEN  GO T
O 60
  160 FOR I=0 TO 93
  165 INK 7
  170 PLOT  INVERSE 1;I,70
  180 DRAW  INVERSE 1;0,24
  190 PLOT  INVERSE 1;190-I,70
  200 DRAW  INVERSE 1;0,24
  205 BEEP 0.01,I/2
  210 NEXT I
1000 REM
1010 REM
1020 REM      MAIN TEXT
1030 REM
1040 REM
1050 LET L=R
1051 LET S=T
1052 LET Q=0
1055 GO SUB MAKE
1060 FOR I=0 TO 2
1070 LET D$=""
```

```
1080 FOR W=LEN C$ TO 1 STEP -1
1090 LET X=INT (RND*W)+1
1100 LET D$=D$+C$(X)
1110 LET C$=C$(1 TO X-1)+C$(X+1
TO )
1120 NEXT W
1130 LET C$="##%&''"
1140 FOR J=1 TO 5
1150 LET E=0
1160 LET F=0
1170 LET G=0
1180 LET H=1
1185 LET N=-1
1186 LET M=CODE D$(J)-25
1190 GO SUB CLEAR
1300 GO SUB MOVE
1310 IF E=1 THEN  GO SUB HOLE: I
F G=1 THEN  GO TO FLOOD
1330 IF E=1 THEN  GO TO 1400
1340 FOR Z=0 TO 5
1350 IF INKEY$="" THEN  LET H=0
1360 IF INKEY$<>"" AND H=0 THEN
 GO SUB NUMBERS
1370 IF F=1 THEN  GO SUB SHOOT:
GO TO 1150
1380 NEXT Z
1390 GO TO 1300
1400 NEXT J
1410 NEXT I
1420 STOP
1500 REM
1510 REM
1520 REM       NUMBERS
1530 REM
1540 REM
1550 LET W$=INKEY$
```

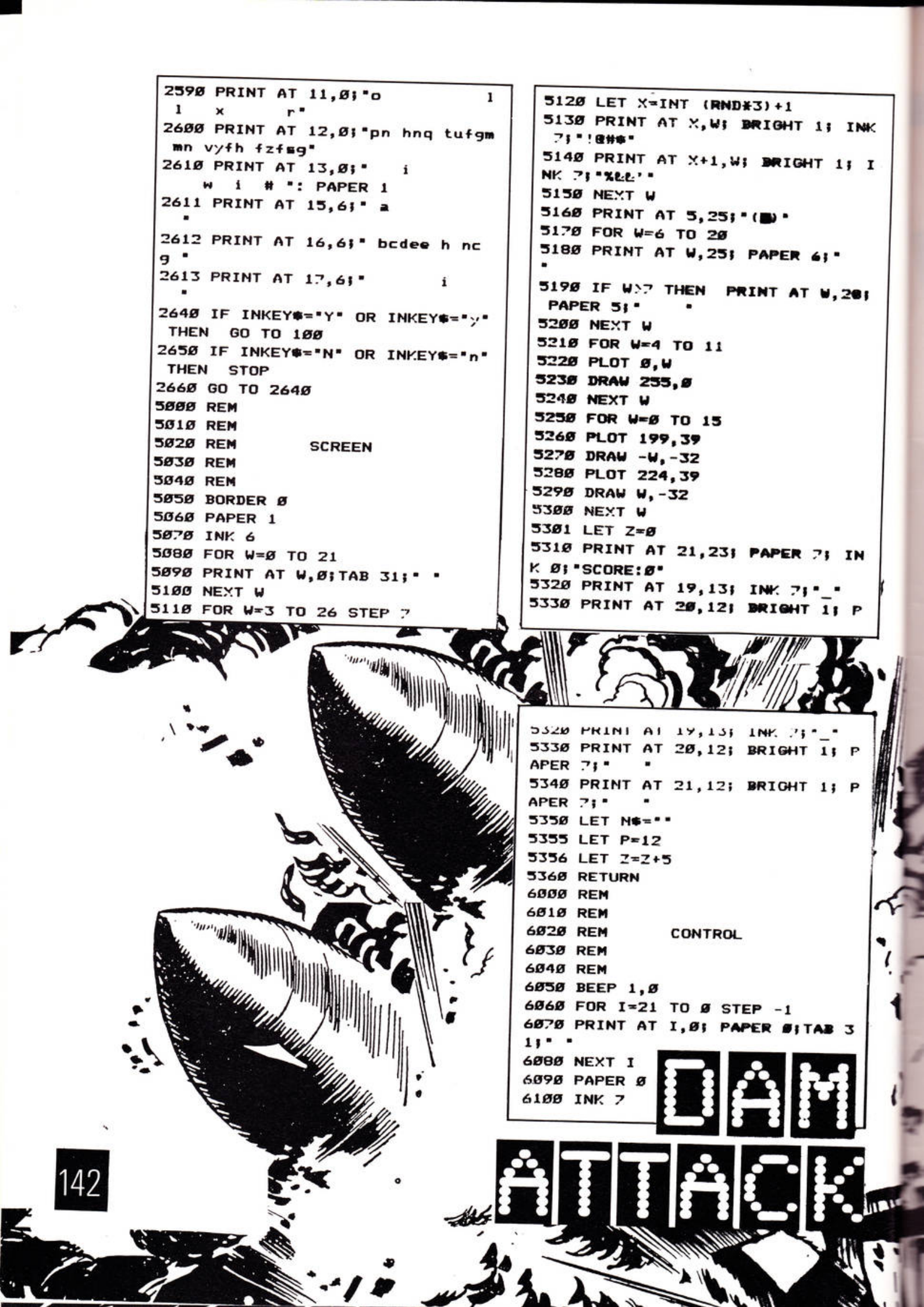


```
1560 IF W$="P" OR W$="p" THEN  G
O SUB CLEAR: RETURN
1565 IF W$=CHR$ 7 THEN  GO TO 60
00
1570 IF W$<"0" OR W$>"9" THEN  R
ETURN
1580 IF P=15 THEN  RETURN
1585 BEEP 0.0005,50
1590 LET N$=N$+W$
1600 LET Y=CODE W$-47
1610 PRINT AT 20,P; BRIGHT 1; PA
PER 7; INK 0;A$(Y)
1620 PRINT AT 21,P; BRIGHT 1; PA
PER 7; INK 0;B$(Y)
1630 IF VAL M$=VAL N$ THEN  LET
E=1
1640 LET P=P+1
1650 LET H=1
1660 RETURN
1700 REM
1710 REM
1720 REM        MOVE SUM
1730 REM
1740 REM
1750 LET N=N+1
1760 IF ATTR (M,N+LEN M$+2)=54 T
HEN  LET E=1
1765 BEEP 0.002,N
1770 PRINT AT M,N;L$
1790 RETURN
1800 REM
1810 REM       MAKE SUM
1820 REM
1830 REM
1832 RANDOMIZE
1835 LET Z=INT ((RND*(L*2))+1)+(
L*3)
1840 IF O$="+" THEN  LET W=INT (
RND*(Z/2))+INT (Z/3): LET X=Z-W:
 IF W>(Z/3)*2 THEN  GO TO 1840
1850 IF O$="-" THEN  LET W=INT (
RND*(Z))+1: LET X=Z+W
1900 LET M$=STR$ X+O$+STR$ W
1910 LET L$=CHR$ 17+CHR$ 1+CHR$
16+CHR$ 0+" "+CHR$ 19+CHR$ 1+"["
+CHR$ 17+CHR$ 0+CHR$ 16+CHR$ 7+M
$+CHR$ 17+CHR$ 1+CHR$ 16+CHR$ 0+
"]"
1990 RETURN
2000 REM
2010 REM
2020 REM         SHOOT
2030 REM
2040 REM
2050 LET X=(100-(N*8))-((1+LEN M
$/2)*8)
2060 LET Y=((19-M)*8)
2065 INK 7
2070 PLOT 100,24
2080 DRAW -X,Y
2090 BEEP 0.4,50
2100 PLOT 100,24
2110 DRAW  OVER 1;-X,Y
2111 IF O=0 THEN  GO TO 2113
2112 LET O=O-1: IF O<1 THEN  LET
 L=L+A: LET O=B: IF L>30 THEN  L
ET L=20
2113 IF K=0 THEN  GO TO 2130
2114 LET K=K-1: IF K<1 THEN  LET
 S=S-C: LET K=B: IF S<1 THEN  LE
T S=1
2130 LET G=G+1
2140 LET V=1
2150 PRINT AT 21,23; PAPER 7; IN
K 0;"SCORE:";G
2155 LET Z=1
2160 FOR X=0 TO 20
2161 LET Z=-Z
2165 INK (6 AND Z=1)+(2 AND Z=-1
)
2170 PRINT AT M,N+1; OVER 1;"\\\
\\\\\";( TO LEN M$+2)
2175 BEEP 0.01,Z
2190 NEXT X
2191 PRINT AT M,N+1; PAPER 1;"
      ";( TO LEN M$+2)
2192 IF V=1 THEN  GO SUB MAKE
2193 FOR X=1 TO 100
2194 NEXT X
2200 RETURN
2300 REM
2310 REM          HOLE
2320 REM
2330 REM
2340 LET V=0
2350 GO SUB 2155
2360 IF I=2 THEN  LET G=1
2370 RETURN
2400 REM
2410 REM
2420 REM         FLOOD
2430 REM
2440 REM
2450 LET Z=21
2460 FOR I=0 TO M
2470 LET Z=Z-1
2475 IF I=Z THEN  PRINT AT I,0;
PAPER 5;"
          ": GO TO 2510
2480 PRINT AT I,20; PAPER 1;"
 "
2490 PRINT AT Z,0; PAPER 5;TAB 2
4;" "
2500 NEXT I
2510 RESTORE 2520
2520 DATA 0.25,-3,0.25,-5,0.15,-
2,0.25,-3,0.45,-5
2530 FOR I=1 TO 5
2540 READ Z,X
2560 BEEP Z,X
2570 NEXT I
2580 INK 7
```

```
2590 PRINT AT 11,0;"o                l
 l   x       r"
2600 PRINT AT 12,0;"pn hnq tufgm
 mn vyfh fzfsg"
2610 PRINT AT 13,0;"    i
    w  i  # ": PAPER 1
2611 PRINT AT 15,6;" a
  "
2612 PRINT AT 16,6;" bcdee h nc
g "
2613 PRINT AT 17,6;"        i
  "
2640 IF INKEY$="Y" OR INKEY$="y"
 THEN  GO TO 100
2650 IF INKEY$="N" OR INKEY$="n"
 THEN  STOP
2660 GO TO 2640
5000 REM
5010 REM
5020 REM         SCREEN
5030 REM
5040 REM
5050 BORDER 0
5060 PAPER 1
5070 INK 6
5080 FOR W=0 TO 21
5090 PRINT AT W,0;TAB 31;" "
5100 NEXT W
5110 FOR W=3 TO 26 STEP 7
```

```
5120 LET X=INT (RND*3)+1
5130 PRINT AT X,W; BRIGHT 1; INK
 7;"!@#$"
5140 PRINT AT X+1,W; BRIGHT 1; I
NK 7;"%&'("
5150 NEXT W
5160 PRINT AT 5,25;"(■)"
5170 FOR W=6 TO 20
5180 PRINT AT W,25; PAPER 6;"
 "
5190 IF W>7 THEN  PRINT AT W,28;
 PAPER 5;"    "
5200 NEXT W
5210 FOR W=4 TO 11
5220 PLOT 0,W
5230 DRAW 255,0
5240 NEXT W
5250 FOR W=0 TO 15
5260 PLOT 199,39
5270 DRAW -W,-32
5280 PLOT 224,39
5290 DRAW W,-32
5300 NEXT W
5301 LET Z=0
5310 PRINT AT 21,23; PAPER 7; IN
K 0;"SCORE:0"
5320 PRINT AT 19,13; INK 7;"_"
5330 PRINT AT 20,12; BRIGHT 1; P
```

```
5320 PRINT AT 19,13; INK 7;"_"
5330 PRINT AT 20,12; BRIGHT 1; P
APER 7;"    "
5340 PRINT AT 21,12; BRIGHT 1; P
APER 7;"    "
5350 LET N$=""
5355 LET P=12
5356 LET Z=Z+5
5360 RETURN
6000 REM
6010 REM
6020 REM         CONTROL
6030 REM
6040 REM
6050 BEEP 1,0
6060 FOR I=21 TO 0 STEP -1
6070 PRINT AT I,0; PAPER 0;TAB 3
1;" "
6080 NEXT I
6090 PAPER 0
6100 INK 7
```

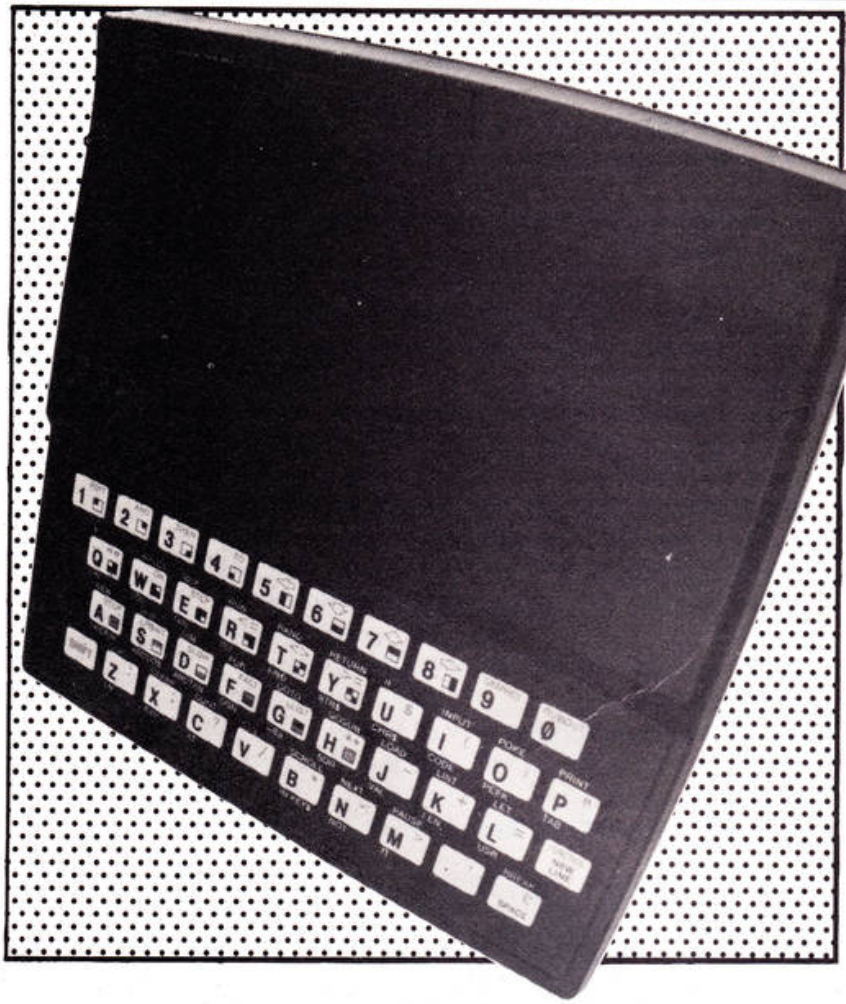


```
6110 PRINT AT 1,13;",  1  x"
6120 PRINT AT 2,13;".ngmcny"
6130 PRINT AT 5,0; PAPER 1; INK
7;" SPEED:1(FAST) - 40(SLOW):  "
;S
6140 PRINT AT 7,0; PAPER 1; INK
7;" LEVEL:1(EASY) - 30(HARD):  "
;L
6150 PRINT AT 9,0; PAPER 1; INK
7;" SUMS :        + OR -      :  "
;O$
6160 PRINT AT 11,0; PAPER 2; INK
 7;" LEVEL INCREASE:";A;" EVERY
";B;" HITS"
6170 PRINT AT 13,0; PAPER 2; INK
 7;" SPEED INCREASE:";C;" EVERY
";D;" HITS"
6180 PRINT AT 17,0; INK 4;" ENTE
R EACH VALUE PRESSING "; PAPER 7
; INK 0;"ENTER"
6190 PRINT AT 18,0; INK 4;"RESPE
CTIVELY"
6200 PRINT AT 5,28; FLASH 1;S
6210 INPUT S
6220 BEEP 0.05,40
6230 IF S<1 OR S>40 THEN  GO TO
6210
6240 PRINT AT 5,28; PAPER 1; INK
 7;S;" "
6250 PRINT AT 7,28; FLASH 1;L
6260 INPUT L
6300 BEEP 0.05,40
6310 IF L<1 OR L>30 THEN  GO TO
6260
6330 PRINT AT 7,28; PAPER 1; INK
 7;L;" "
6340 PRINT AT 9,28; FLASH 1;O$
6350 INPUT O$
6360 BEEP 0.05,40
6370 IF O$<>"+" AND O$<>"-" THEN
  GO TO 6350
6380 PRINT AT 9,28; PAPER 1; INK
 7;O$
6390 PRINT AT 11,0; PAPER 2; INK
 7;"LEVEL INCREASE:"; FLASH 1;A;
 FLASH 0;"EVERY ";B;" HITS"
```

```
6400 INPUT A
6410 BEEP 0.05,40
8420 IF A<0 THEN  GO TO 6400
8430 PRINT AT 11,0; PAPER 2; INK
 7;" LEVEL INCREASE:";A;" EVERY
";B;" HITS"
8450 INPUT B
8460 BEEP 0.05,40
8470 IF B<0 THEN  GO TO 8450
8475 LET O=B
8480 PRINT AT 11,0; PAPER 2; INK
 7;" LEVEL INCREASE:";A;" EVERY
";B;" HITS"
8500 INPUT C
8510 BEEP 0.05,40
8520 IF C<0 THEN  GO TO 8500
8530 PRINT AT 13,0; PAPER 2; INK
 7;" SPEED INCREASE:";C;" EVERY
";D;" HITS"
8550 INPUT D
8560 BEEP 0.05,40
8570 IF D<0 THEN  GO TO 8550
8575 LET K=D
8580 PRINT AT 13,0; PAPER 2; INK
 7;" SPEED INCREASE:";C;" EVERY
";D;" HITS"
8590 PRINT AT 17,0; INK 7;" TO R
ETURN THIS PAGE PRESS EDIT
                           .
8600 FOR Z=1 TO 200: NEXT Z
8601 LET R=L
8602 LET T=S
8610 GO TO 90
8999 STOP
9000 REM
9010 REM
9020 REM      VARIABLES
9030 REM
9040 REM
9050 LET A=0
9060 LET B=0
9070 LET C=0
9080 LET D=0
9090 LET L=10
9100 LET S=20
9110 LET A$="SACEGIKMOO"
9120 LET B$="TBDFHJLNPR"
9130 LET C$="#$%&'"
9140 LET O$="+"
9150 LET SCREEN=5000
9160 LET CLEAR=5330
9170 LET NUMBERS=1500
9180 LET MOVE=1700
9190 LET MAKE=1800
9200 LET SHOOT=2000
9210 LET HOLE=2300
9220 LET FLOOD=2400
9230 RETURN
9998 REM
9999 REM     END OF PROGRAM
```

CODE BREAKER

BY NORMAN STEVENS



Can you beat the computer at a game of logic? Try it out with your 16K ZX81.

In this version of Codebreaker, you challenge your ZX81 (which is pretty good, be warned!). For those of you who have not actually seen this type of game in action, the object is to guess a four-digit number by trial and error (although some may like to suggest that a great deal of skill is involved). After each guess you are told (or you tell the computer) how many digits are correct and how many are in the correct position. Running totals and averages are printed to let you know how well you are doing.

Program breakdown

Line 1	Machine code routine to evaluate the ZX81's next attempt
Lines 10-70	Main routine for controlling game
Lines 100-240	Subroutine to initialise global variables. C$ and A$ must be the first variables declared otherwise the machine code routine won't work
Lines 300-420	Initialises variables for ZX turn
Lines 500-650	ZX turn
Lines 700-760	Initialising variables for player's turn
Lines 800-990	Player's turn
Lines 1000-1160	Subroutine to print the ZX81's latest attempt and to receive input (CORR and INPOS)
Lines 1400-1440	Calculates four-digit random number
Lines 1500-1730	Error handling. The machine code program sends you here if it can't find a solution that fits all responses to previous attempts. Returns to state of play before incorrect response given

Variables used

A$(12,6)	Store for keeping ZX81's attempts
C$(4)	Current ZX81 attempt/solution for your attempt
A(2)	Number of games played by each player
B(2)	Total number of attempts by each player
C(2)	Averages
D$(4)	Player's current attempt
E$	Player names
PL	Player number: 1=ZX81; 2=player
GOES	Number of attempts in current game
CORR	Number of digits correct in current attempt
INPOS	Number of digits in correct position in current attempt

```
  1 REM E(RND)▪ ;2£ FOR ; ▪ ,ACS
▒▒(74▪ £( PRINT STR$ FAST ▪ ▪ LP
RINT FAST VAL ▪▪ ,▒4▪ACS RETURN A
T £/▪ 7( POKE AT ,(( REM LPRINT 77
777/▒Y9 LPRINT ▪▪ O▒K▒007( RUN 5)
=▪ ?▪:
 10 REM INITIALISE GLOBAL VARS
 11 GOSUB 100
 20 IF PL=2 THEN GOTO 50
 30 REM INITIALISE VARS FOR ZX
 31 GOSUB 300
 40 REM ZX TURN
 41 GOSUB 500
 50 REM INITIALISE VARS FOR
PLAYER TURN
 51 GOSUB 700
 60 REM PLAYER TURN
 61 GOSUB 800
 70 GOTO 30
 99 REM
100 REM INITIALISE GLOBAL VARS
101 REM
109 REM THIS REQUIRED BY MC
ROUTINE
110 CLEAR
111 DIM C$(4)
112 DIM A$(12,6)
113 REM
120 DIM A(2)
130 DIM B(2)
140 DIM C(2)
150 DIM D$(4)
160 DIM E$(2,4)
170 LET E$(1)="MY"
180 LET E$(2)="YOUR"
190 PRINT "MASTERMIND CHALLENGE
"
200 PRINT "DO YOU WANT TO GO FI
RST?"
210 INPUT 0$
220 LET PL=(CODE 0$=51)+2*(CODE
 0$=52)
230 IF PL=0 THEN GOTO 210
240 RETURN
299 REM
300 REM INITIALISE VARS FOR ZX
301 REM
310 RAND
320 LET SP=1
330 LET GOES=0
340 SCROLL
350 PRINT "HAVE YOU THOUGHT OF
A NUMBER?"
360 SCROLL
370 PRINT "PRESS ANY KEY TO CON
TINUE"
380 IF INKEY$="" THEN GOTO 380
390 FOR I=SP TO 12
400 LET A$(I)=""
410 NEXT I
420 RETURN
499 REM
500 REM ZX FIRST TWO GOES
501 REM
510 FOR I=SP TO 2
520 REM GET RANDOM C$
521 GOSUB 1400
530 REM ZX TURN INQUIRY
531 GOSUB 1000
540 IF INPOS=4 THEN GOTO 1200
550 NEXT I
599 REM
600 REM FURTHER ZX GOES
601 REM
610 LET C$="0000"
620 LET ZZ=USR 16514
630 REM ZX TURN INQUIRY
631 GOSUB 1000
640 IF INPOS=4 THEN GOTO 1200
650 GOTO 820
699 REM
700 REM INIT VARS FOR PLAYER
701 REM
710 RAND
720 LET GOES=0
730 REM GET RANDOM C$
731 GOSUB 1400
740 SCROLL
750 PRINT "OK I""M READY"
760 RETURN
799 REM
800 REM PLAYER TURN
801 REM
810 LET B$=C$
820 INPUT D$
830 SCROLL
840 PRINT D$;" ";
850 LET CORR=0
860 LET INPOS=0
870 FOR I=1 TO 4
880 IF D$(I)<"0" OR D$(I)>"9" T
HEN GOTO 820
890 IF D$(I)=C$(I) THEN LET INP
OS=INPOS+1
900 FOR J=1 TO 4
910 IF D$(I)=B$(J) THEN GOTO 94
0
920 NEXT J
930 GOTO 960
940 LET CORR=CORR+1
950 LET B$(J)=""
960 NEXT I
970 LET GOES=GOES+1
980 PRINT CORR;" CORRECT ";INPO
S;" IN POSITION"
990 GOTO 800+400*(INPOS=4)
999 REM
1000 REM INQUIRY ZX TURN
1001 REM
1010 SCROLL
1020 PRINT C$;" ? CORRECT"
1030 INPUT CORR
1040 LET CORR=INT CORR
1050 IF CORR<0 OR CORR>4 THEN GO
TO 1030
1060 PRINT AT 21,5;CORR;TAB 15;"
? IN POSITION"
1070 INPUT INPOS
1080 LET INPOS=INT INPOS
1090 IF INPOS<0 OR INPOS>CORR TH
EN GOTO 1070
1100 PRINT AT 21,15;INPOS;TAB 29
;"OK?"
1110 INPUT 0$
1120 IF CODE 0$=51 THEN GOTO 101
0
1130 LET A$(SP)=C$+CHR$ INPOS+CH
R$ CORR
1140 LET SP=SP+1
1150 LET GOES=GOES+1
1160 RETURN
1199 REM
1200 REM PRINT SCORE ETC.
1201 REM
1210 SCROLL
1220 PRINT "GOT IT IN ";GOES
1230 SCROLL
1240 LET A(PL)=A(PL)+1
1250 LET B(PL)=B(PL)+GOES
1260 LET C(PL)=INT (B(PL)*100/A(
PL)+0.5)/100
1270 PRINT A(PL);" GAMES, ";E$(P
L, TO 2*PL);" AVGE IS ";C(PL)
1280 SCROLL
1290 IF C(1)>C(2) THEN PRINT "YO
U""RE AHEAD (AT THE MOMENT)"
1300 IF C(1)=C(2) THEN PRINT "WE
 ARE ABOUT LEVEL"
1310 IF C(1)<C(2) THEN PRINT "I"
"M WINNING"
1320 LET PL=3-PL
1330 RETURN
1399 REM
1400 REM GET RANDOM C$
1401 REM
```

CODE BREAKER

```
1410 LET B$=STR$ RND
1420 IF LEN B$<6 THEN GOTO 1410
1430 LET C$=B$(3 TO )
1440 RETURN
1499 REM
1500 REM ERROR HANDLING
1510 SCROLL
1520 PRINT "I THINK YOU""VE MADE
 A MISTAKE"
1530 SCROLL
1540 PRINT "LET""S TRY AGAIN"
1550 SCROLL
1560 SCROLL
1570 PRINT "WHICH LINE IS WRONG"
1580 INPUT ERR
1590 LET ERR=INT ERR
1600 IF ERR<0 OR ERR>=SP THEN GO
TO 1550
1610 REM RESTORE + PRINT CORRECT
 TRIES
1620 CLS
1630 FOR I=1 TO ERR-1
1640 FOR J=1 TO 4
1650 IF CODE A$(I,J)>127 THEN LE
T A$(I,J)=CHR$ (CODE A$(I,J)-128
)
1660 NEXT J
1670 SCROLL
1680 PRINT A$(I, TO 4);" ";CODE
A$(I,6);" CORRECT ";CODE A$(I,5)
;" IN POSITION"
1690 NEXT I
1700 LET SP=ERR
1710 LET GOES=ERR-1
1720 GOSUB 390
1730 GOTO 500
8000 FOR I=16514 TO 16609 STEP 5
8010 LET SUM=0
8020 LPRINT I;" ";
8030 FOR J=0 TO 4
8040 LET L=PEEK (I+J)
8050 LPRINT L;",";
8060 LET SUM=SUM+L
8070 NEXT J
8080 LPRINT " ";SUM
```

```
8090 NEXT I
8100 STOP
9000 IF PEEK 16511+256*PEEK 1651
2<102 THEN PRINT "REM LINE TOO S
HORT";ERR
9010 FOR I=16514 TO 16609 STEP 5
9020 LET SUM=0
9030 SCROLL
9040 PRINT I;" ";
9050 FOR J=0 TO 4
9060 INPUT L
9070 PRINT L;",";
9080 LET SUM=SUM+L
9090 POKE I+J,L
9100 NEXT J
9110 INPUT L
9120 PRINT " ";L
9130 IF SUM=L THEN GOTO 9170
9140 SCROLL
9150 PRINT "ERROR IN LINE"
9160 GOTO 9020
9170 NEXT I
```

Spectrum Computing adds a new dimension to your micro!

Run this Argus Spectrum tape and you'll soon see why it's Britain's top selling tape magazine. Each issue gives you a variety of exciting and challenging games to play, reviews of other newly released software plus valuable utilities enabling you to write your own programmes and games.

Stretch your imagination and skills with Spectrum Computing – available every other month from WH Smith, Menzies and other leading stores.

(You'll see them advertised on TV from September!)

Get your copy today!

Argus Press Software Group

Argus Tape Magazines produced by
ARGUS PRESS SOFTWARE
1 Golden Square, London W1R 3AB
Telephone: 01 437 0626

A L I E N

In space no one can hear you scream.

NAVIGATOR
Shy, Skilful and Intelligent — Panics Easily.

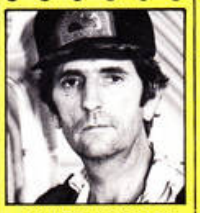
EXECUTIVE OFFICER
Direct, Imaginative, Cautious, Loyal.

SCIENCE OFFICER
Secretive, Unlikeable, Brilliant — Occasionally Illogical.

CAPTAIN
Solid, Dependable, Courageous — Excellent Leader.

ENGINEERING OFFICER
Physically Strong, Low I.Q. Potentially Rebellious

3RD OFFICER
Wilful, Ambitious, Authoratitive, Resourceful.



ENGINEERING OFFICER
Cynical, Rebellious, Untrustworthy, Unflappable.

THE CRE
Personnel files follow - yours to command - well almost.

MIND GAMES
SPECTRUM 48K · CBM64

Argus Press Software Group

Featuring the unique Personality Control System

£8.99

No. 1 Golden Square, London W1R 3AB, Telephone 01-437 0626