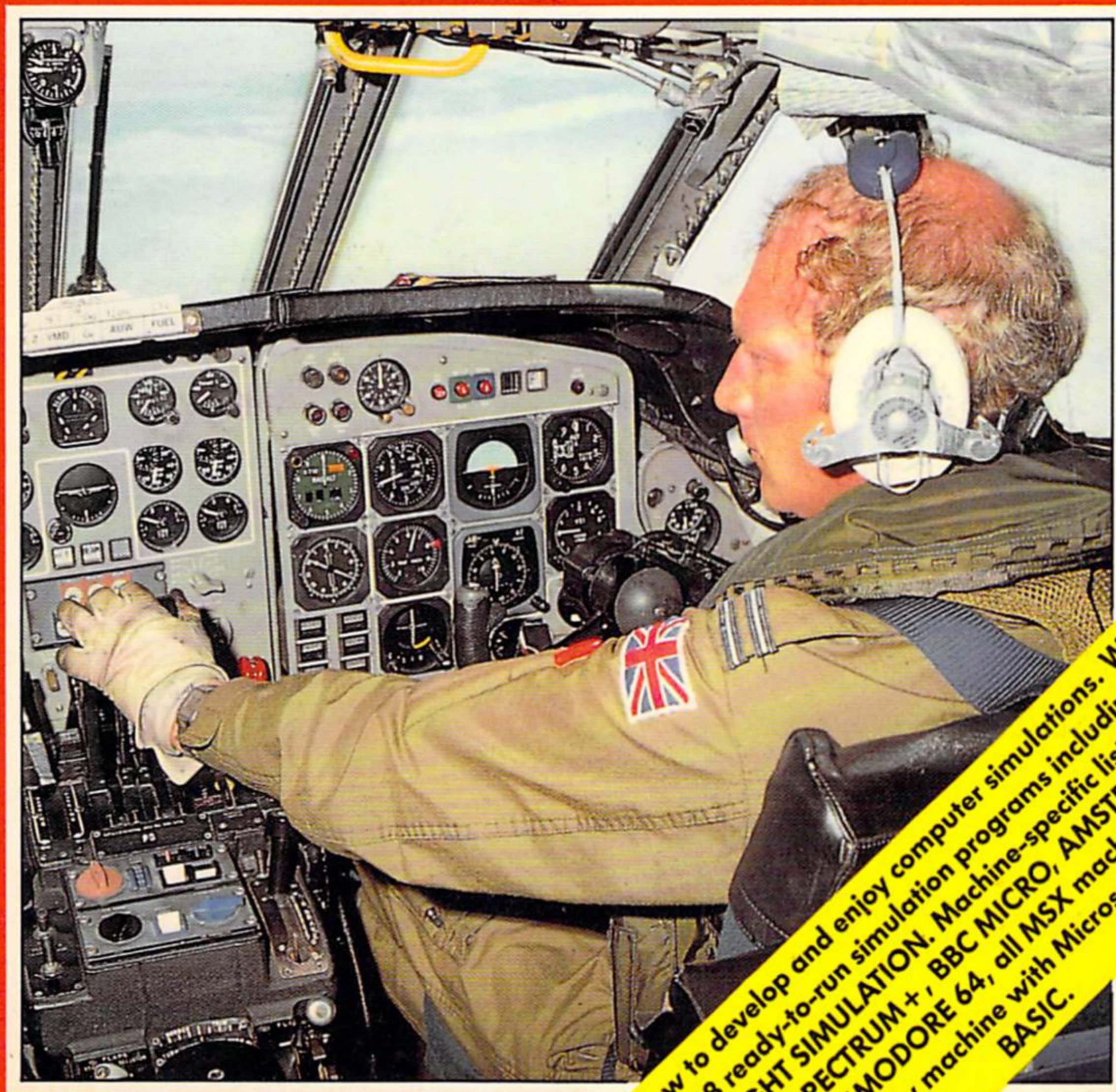
REPLICATING REALITY:

EXPLORING COMPUTER SIMULATIONS

Tim Hartnell



How to develop and enjoy computer simulations. With 18 ready-to-run simulation programs including FLIGHT SIMULATION. Machine-specific listings for SPECTRUM+, BBC MICRO, AMSTRAD, COMMODORE 64, all MSX machines, any machine with Microsoft BASIC.

Interface Publications
LONDON · MELBOURNE

REPLICATING REALITY —

Exploring Computer Simulations

Tim Hartnell

Listings provided for the following computers:

Amstrad
BBC Micro
Spectrum+/Spectrum
Commodore 64
Apple II
Dragon
Atari
All MSX machines
Plus any computer furnished with Microsoft BASIC

First published in the UK by:
Interface Publications Ltd.,
9-11 Kensington High Street,
LONDON W8 5NP

Copyright © Tim Hartnell, 1985
First printing September, 1985

ISBN 0 907563 80 5

The programs in this book have been included for their instructional value. They have been tested with care, but are not guaranteed for any particular purpose. While every care has been taken, the publishers cannot be held responsible for any running mistakes which may occur.

ALL RIGHTS RESERVED

No use whatsoever may be made of the contents of this volume - programs and/or text - except for private study by the purchaser of this volume, without the prior written permission of the copyright holder.
Reproduction in any form or for any purpose is forbidden.

Books published by Interface Publications are distributed in the UK by WHS Distributors, St. John's House, East Street, Leicester LE1 6NE (0533 551196) and in Australia and New Zealand by PITMAN PUBLISHING. Any queries regarding the contents of this volume should be directed by mail to Interface Publications, 9-11 Kensington High Street, London W8 5NP.

Printed and Bound by Short Run Press Ltd, Exeter

CONTENTS

Author's introduction vii
Getting the programs running viii
Chapter One – The Power of Simulations 1
Chapter Two – Components of Simulation 7
Chapter Three – Feedback in Action 15
Chapter Four – Monte Carlo Simulations 21
Chapter Five – Simultaneous Equations 27
Chapter Six – The Meaning of Life 35
Chapter Seven – Robot Simulations 45
Chapter Eight – The *Robot Logo* Listing 55
Chapter Nine – The Point-Duty Robot 61
Chapter Ten – Simulating Intelligence 75
Chapter Eleven – The Quevedo Chess Machine 81
Chapter Twelve – Into the Political Arena 91
Chapter Thirteen – Playing the Stock Market 103
Chapter Fourteen – The *Stock Market* Listing 109
Chapter Fifteen – Running a Vehicle Factory 115
Chapter Sixteen – The *Detroit City* Listing 123
Chapter Seventeen – Life at the Super Bowl 129
Chapter Eighteen – The *Gridiron* Listing 135
Chapter Nineteen – The Grand Slam 143
Chapter Twenty – The *Tennis* Listing 149
Chapter Twenty-One – Driving a Racing Car 155
Chapter Twenty-Two – The *Grand Prix* Listing 161
Chapter Twenty-Three – Up, Up and Away 169
Chapter Twenty-Four – The *Flight Simulation* Listing 179
Listings for Commodore 64 187
Listings for BBC Micro 260
Listings for Spectrum, Spectrum + ... 320
Further Reading 363
Acknowledgements 365

AUTHOR'S INTRODUCTION

This book will give you and your computer power. Power to replicate reality in a way which can be quite uncanny. From microbes which fight it out in a closed environment for the right to survive, to a program which puts you in the cockpit of a particularly temperamental plane, there is a wide range of programs to allow you to experience this power in action.

There is more than programs in this book. As well as the ready-to-run material, you'll discover how easy it is to analyse real-life situations so you can create your own computer simulations, for entertainment and instruction.

You now have the power. Use it.

Tim Hartnell,
London, 1985

Tim Hartnell is one of the world's top-selling computer authors. Recent works include *Exploring Artificial Intelligence on your Commodore 64* (Bantam Books, New York, 1985; Interface Publications Ltd., London, 1985), *Creating Adventure Programs on your Microcomputer,* (Ballantine, New York, 1984) and *The Personal Computer Handbook* (Virgin Books, London, 1984). He is founding editor of the magazine *ZX Computing,* and contributes a regular column *(Response Frame)* to *Your Computer* magazine.

GETTING THE PROGRAMS RUNNING

The programs were all written originally on an IBM PC. The listings in the main body of the book are from the IBM PC and will run, without modification, on all the IBM PC, IBM XT and compatible machines. The listings have deliberately been written in a common subset of Microsoft BASIC, and, with one exception, will run with minimum changes on any machine fitted with BASIC. The exception is the POINT-DUTY ROBOT program, which demands access to high-resolution graphics, and the use of a command such as PSET or POINT. You may need to consult your system's manual to find out the equivalent command for your machine.

To seed the random number generator, I have used the line:

```
RANDOMIZE VAL(RIGHT$(TIME$,2))
```

which you should replace with your own system's equivalent (or leave out altogether, if you do not need such a line).

Specific versions of the programs for the Spectrum/Spectrum+ and for the Commodore 64 appear near the end of the book. If you have an Amstrad or a BBC Micro use the program listings as they are in the main body of the book.

Chapter One –
The Power of Simulations

Simulations give you power. Power to try your hand vicariously at almost any activity you can imagine. Running a multi-million dollar manufacturing concern, flying to the moon, driving a racing car or running the country as president – all these experiences can be yours through computer simulations.

In this book, you'll find programs to allow you to take part in all these scenarios, and more. As well, you'll be gently introduced to the art of creating computer simulations, so that after you've enjoyed the programs in the book, you can write your own. Studying the listings in this book, observing the way the variables interact, and mathematical formulae are used to replicate 'real life' situations, will give you a number of easy-to-apply techniques to use in your own simulations.

Despite their entertainment value, computer simulations are not just elaborate games. Simulations have found ready acceptance in many situations where reality is too expensive, too dangerous or too complex to allow human beings to learn to control. Many, many hours of airtime during pilot training are clocked up in elaborate flight simulators – controlled, of course, by computers – rather than up in the air.

Operators in conventional and nuclear power-plants around the world train on simulators, where they can learn their skills without facing the real catastrophes which could come from making mistakes. One example of the use of computer-controlled simulations in this area comes from Australia, where an electricity generating board has spent half a million dollars to buy a simulator to train its operators. The simulator, developed and marketed by an Australian company called Control Simulation Technology,

allows a number of emergency situations to be created, so that staff can learn to cope with them. Such training would be impossible without the use of a simulator.

Another computer simulator, which might be seen as perhaps a little sinister, uses the reactions of a few test subjects to simulate the reaction of a significant portion of the population in testing the effectiveness of television and print advertising. The simulator, developed by a psychologist named Marcus Tomlian, is claimed to be "the most sophisticated precision system for measuring consumer reactions to advertisements". The device, called Mind Monitor, is made up of a series of sensors for taking brain and heart readings of the sample group of people. The readings from these sensors are fed into the simulator proper, which analyses the brain waves and cardiovascular readings, and from them extrapolates the probable reaction of the population as a whole. The simulator runs while an advertisement is shown to the test group, and gives a second by second reaction readout.

Mr. Tomlian says Mind Monitor can therefore be used to find weak points in an advertising campaign, leading to 'fine-tuning' of particular campaign elements. General Electric is believed to be currently at the forefront of developing computer simulations which can help gauge viewers' attention to advertising, while one of the world's biggest agency networks, Yound and Rubicam, is also investigating the field.

Whether you wish to replicate the chemical processes inside an atomic reactor, mimic the reactions of potential purchasers to consumer advertising, or imitate the control system of the Space Shuttle, a computer simulation can be employed. With a simulation, the reactions and actions of a complex system can be represented in quite graphic detail. Often, the simulation can be manipulated in ways which would be difficult, dangerous or just impossible with the real situation. For example, a pilot training on a flight simulator could attempt to land during a hurricane with very little fuel, or an operator at a nuclear power station could be faced with the possibility of a melt-down.

Limitations

A computer simulation, of course, must never be confused with the system it is simulating. All simulations are, to a greater or lesser extent, *simplifications* of the system they represent. Low level simulations, like the programs in this book, and the simulations you are likely to create in the next few months, are often drastic simplifications. Despite this, they can be surprisingly effective, and trying to land using our *Flight Simulation* program can be almost as nerve-racking as the real thing.

You'll find that there is as much art as science in creating effective simulations. The science is working out the formulae to use which link the interacting elements of the system you are attempting to simulate. The art is presenting those formulae in a way which makes the experience of interacting with the simulation as 'life like' and rewarding as possible. This means the output must be realistic, and must be in a form which is readily understood. With many simulations, the speed with which the program responds to user input is very important. If the simulation is occuring in 'real time' (that is, a second of time elapsed within the actions of the simulation is intended to equal a second of real-life time) it is important that the simulation is written so that this one-to-one correspondence between the two times is adhered to as closely as possible.

Simulations can be divided into two groups – those where the data comes in separate, discontinuous packages, and those where the data values flow into each other. Queuing systems, where elements to be serviced (such as customers at a bank counter) wait in a line until a free service point is available, are the predominant type of discontinuous situations which are often examined using simulations. In the second type of simulation, where the data values flow into each other, values (such as the internal heat of a nuclear reactor) are constantly changing with respect to time.

It is important to try and determine, before you start any programming, which data type is predominant within the real-world system you are going to program.

Any situation where elements are added to a queue at rate A and removed from the other end at rate B is a candidate for a discrete simulation. There are many, many such situations in real life, other than simply waiting to be served in a line at the supermarket checkout. Discrete queues exist in such disparate situations as egg collection in an old-fashioned farmyard, when a queue can be seen in the difference in time between eggs being laid and gathered), road accident victims waiting for ambulances, and airplanes at JFK waiting for a take-off window.

In this book, we have a *Space Landing Simulation* in which you have to try and land a space craft on the moon. This program waits for an input (a number, representing 'thrust', between zero and nine) before continuing. If the program did not wait for a key press, but kept going through its operational cycle over and over again, strobing the keyboard and making changes *if* it detected a valid key press, the simulation would clearly be a continuous one. There would be no instant when the velocity and position of the craft were not changing. Add a similar 'real-time' control facility to our *Flight Simulation* program, and you'd have another example of a non-discrete simulation.

The predator/prey simulation, in the chapter on simultaneous equations, is also a continuous one. There is no point in time, once the program begins cycling through the major loop, when the values of the two major variables (the 'predator' and the 'prey') are not being updated.

In the next chapter of this book, we'll look at the basic elements of a simulation, and show how a problem can be broken down so that a simulation of it can be created. The subsequent chapters explore many types of simulations, so you can see various simulation techniques in action, and can study the programs to see how the results were achieved. The lessons you learn from these chapters can then be applied to help you create your own simulations.

Chapter three looks at loops and feedback, the fundamental elements of all simulations, and shows these in action with the *Space Landing Simulation* program. From there we move to Monte Carlo Simulations, where the effects of random inputs on a system are discussed. Chapter five is the 'predator/prey' simulation mentioned earlier, where simultaneous equations model the

interaction of two kinds of cells, living within a culture medium, and competing for the right to survive.

The next chapter presents John Conway's classic computer simulation, *Life,* where birth and death occur in a less blood-thirsty way than in our 'predator/prey' simulation. The initial version of this program allows you to encode starting generations as DATA statements, so evolutionary patterns can be studied several times. Then you're shown the simple changes to make in order to allow your computer to generate the starting colony at random.

Chapters seven, eight and nine – which look at robot simulations – are considerably more ambitious than the preceding ones. In these chapters, we will actually examine two simulations at once. Firstly, we emulate the action of a computer language interpreter, which 'understands' commands entered in a new language RCL ('Robot Control Language'). Secondly, the programs use their understanding of RCL to move a little robot around the screen, which leaves Logo-like designs as it trundles around. The programs given here form a solid basis for developing a complete turtle graphics program.

Artificial intelligence is one of the 'hot' areas of computer science at the moment, the book would not be complete without a program (or two) which showed how a simulation program can apparently endow a computer with intelligence. Chapter ten allows you to attempt the fairly fruitless task of trying to beat your computer at the game *Connect Four,* and explains how the program assesses its 'best' move. The move-selection technique can be adapted for many board games. The following chapter, eleven, introduces a program which replicates an 1890 Spanish machine which plays a particular chess end game. The Quevedo Chess Machine was probably the first genuine attempt to produce a chess-playing machine, so it is appropriate that we have it replicated in this book.

From chess we move into the political arena, and show how the economy of the US can be (admittedly very crudely) modelled within a computer simulation. You are the president, and you have to try to control the economy. You may well have a deeper sympathy for the real occupant of the Oval Office after you find your best efforts blowing up in your face.

Money manipulation lies at the heart of chapter thirteen's simulation of a small stock market in action, as you invest in the fluctuating fortunes of five companies trying to reach a financial goal you have set for yourself. If you find you have considerable skill in this area, you can see if it will stand you in good stead in manufacturing. *Detroit City,* in chapters fifteen and sixteen, confronts you with a difficult challenge, as you try to bring an ailing automobile manufacturing company back to profitable life. Fail, and you're out of a job. Succeed, and you get a seat on the board.

We move from the world of business simulations to programs which simulated leisure activities. *Gridiron,* in chapters seventeen and eighteen, puts you up against the computer-controlled *Silicon Cowboys* (or you can use the program to moderate a clash between two human beings). The next program gives you the chance to play a three-set game of tennis against your computer. Again, if you prefer, you can use the simulation simply to moderate the action, and play against another flesh-and-blood opponent.

In chapters twenty-one and twenty-two, you are at the wheel of a racing car, as you try out your skill on your choice of Grand Prix track, in Britain, Germany, Italy or Monaco. The tracks within the program are based on the actual tracks in real life, although I doubt whether the skills you pick up here could very usefully be transferred to genuine Grand Prix driving on those circuits.

Finally, in chapters twenty-three and twenty-four, we have our *simulation de resistance,* a complete flight simulator. The program simulates an airplane which is fiendishly difficult (but not impossible) to fly. You'll find that keeping track of four or five constantly-changing factors at once will take all your concentration. But the satisfaction you'll feel on your first successful landing will make it all worthwhile.

As you can see, we have some fascinating territory ahead of us. Let's go.

Chapter Two
Components of A Simulation

A simulation is a computer model of cause and effect. Event A is linked to event B by equation X. Modify B and C is affected, by linking factor Y. And so on. Isolate the links (X and Y, in the preceding lines) and you have the raw ingredients of a simulation.

If you turn on a light switch, electricity will flow through the circuit, and the light will come on. Cause and effect are easy to see in this situation:

```
CAUSE ---------> EFFECT ---------> EFFECT
SWITCH ON        CURRENT FLOWS     LIGHT GLOWS
```

It would be, of course, a trivial exercise to write a program to simulate the result of turning on a light switch. However, this should not blind us to the fact that, in essence, all simulations you write will be a clear linking between causes and effects.

More Complex Links

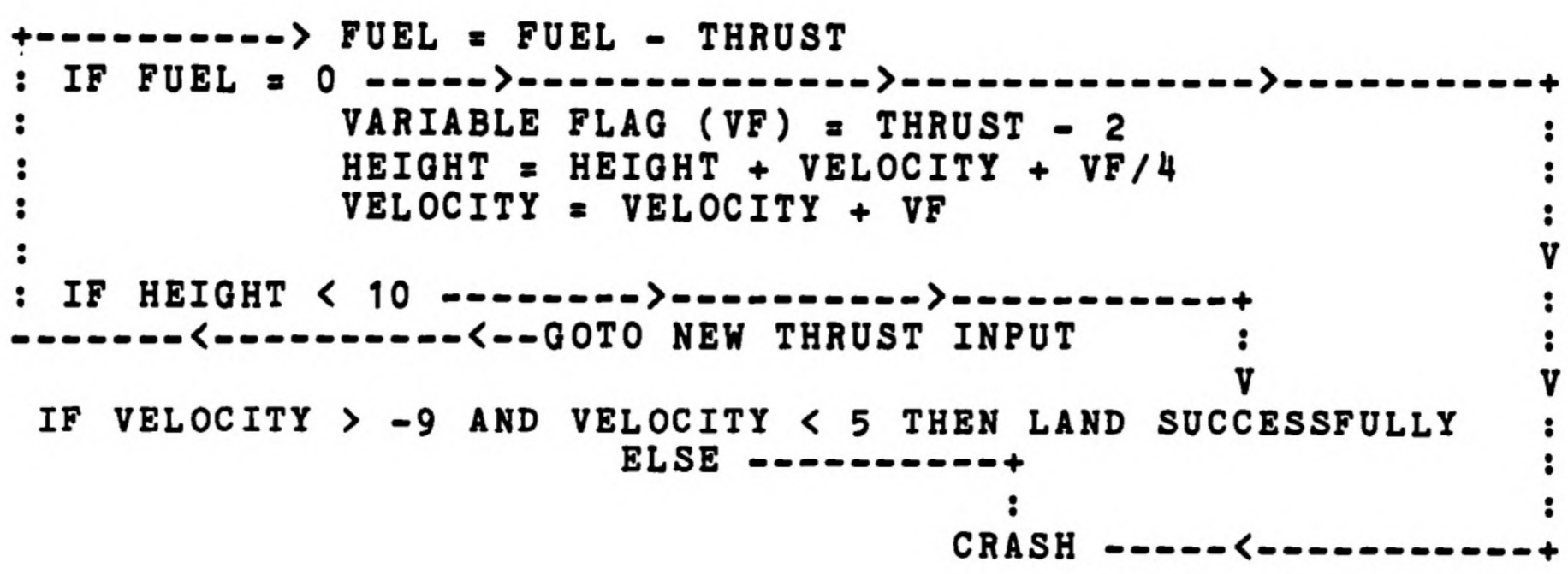
The links may not, of course, be as simple and straightforward. Some effects may only come into play when variables reach 'trigger values' (such as the car engine in our *Grand Prix* simulation blowing up only if the engine temperature exceeds 200 degrees). Your own simulations may have a link like this:

```
EVENT X -->--+-----leads, if A present to --> EVENT Y
             :
             +-----leads, if B present to --> OUTCOME Z
```

Analyse the links, and express them in a diagram like the above (perhaps substituting mathematical formulae for lines like 'leads, if A present to') and you're well on the way to writing your own simulation program.

The first simulation program in this book (in the next chapter) puts you in the position of a pilot trying to land your space craft on the moon. The causal links in this program are reasonably straightforward, but still manage to produce a very interesting (and somewhat unpredictable) outcome.

Here are the links (and THRUST is the input from the user, each time through the loop):



Although this may look a little complex at first sight, it is actually fairly simple. You can see that the causal loop diagram looks pretty much like a flow chart, and actually contains some lines which could apparently be programmed directly. When you look at the program (especially lines 17Ø through to 24Ø) you'll see that some of the lines above do appear almost unchanged in the program.

Feedback

Causal loops contain elements which interact. They also contain feedback. The results of one series of computations (in the moon-lander simulation, the results are the height, velocity and fuel left) feed back to the user to affect the next input (THRUST in this case). The feedback in this case comes from outside the system, from the user.

Feedback can be seen in many situations, such as that involving a thermostat on a room heater. If the room gets colder, the heater is turned up. When the room becomes warmer, the heater is turned down. In due course, the room temperature drops, and the cycle begins again.

Feedback does not always lead to a fluctuating output. I am angry with you. I raise my voice. You shout back. I raise my voice some more, as I am further angered by you raising your voice. You, in turn, are made more angry by my increased volume, and shout even more loudly at me . . . and a self-reinforcing loop is set up. If the heater's thermostat was set so that the hotter the room became, the more heat was produced, a similar self-reinforcing feedback loop would be seen (and felt!).

Equilibrium Points

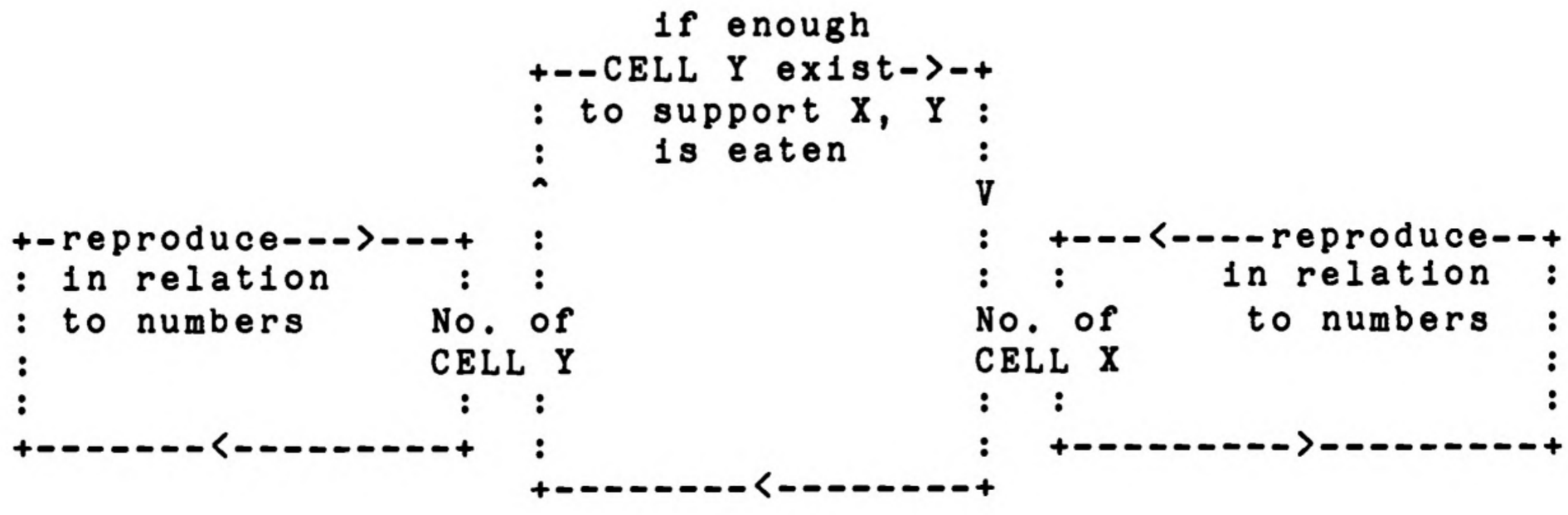
Many feedback loops have an *equilibrium point,* and a good simulation may well be, in effect, inviting the program user to discover that equilibrium point. There may well be, for example, a thrust input with our moon-lander which allows the craft to sink gently down into the luna dust.

Negative and Positive Loops

A *negative* feedback loop (as the room gets hotter, the thermostat turns the heater down) tends to push the system towards an equilibrium point. A *positive* feedback loop (as the room gets hotter, the heater is turned up more) tends to lead to dramatically accelerating (or decelerating) growth rates, pushing the system away from an equilibrium point.

It is fairly easy to understand how a single feedback loop can form the heart of a simple simulation. However, most situations in real life, when analysed, turn out to be made of more than one feedback loop.

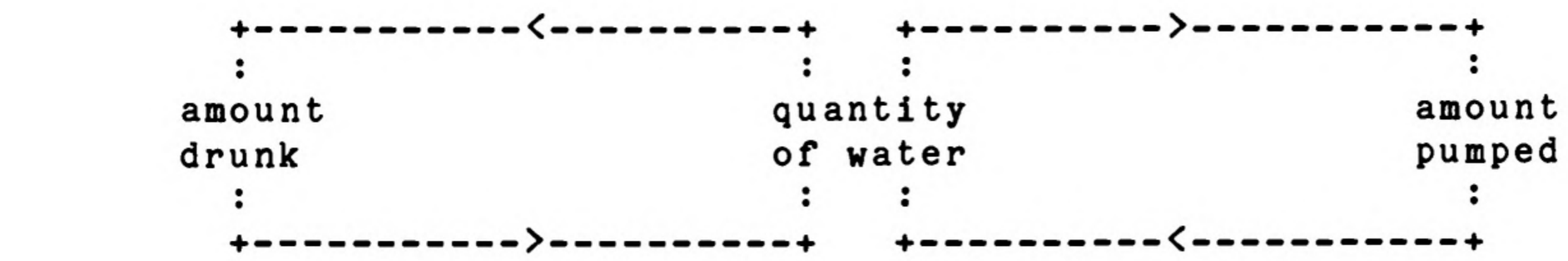
For example, in the 'predator/prey' simulation (chapter five), we have two life forms, whose survival is inextricably bound together. If one species (the prey) dies out, the other will also die. If there are few predators, the prey numbers increase, leading to more food, so more predators are born, and these in turn eat more prey. The situation is dynamic, and constantly changing, as the simulation gives feedback to itself, which modifies its next run. Here is how chapter five's program looks in a causal diagram:



A situation like this can stabilise, or move so that either X or Y becomes dominant, with the other cell reduced to an ineffectual level. Such simulations are fascinating to run, and their outcomes can be difficult to predict, even when you know the linking formulae well. In a simulation like this, where there are two negative feedback loops in operation, X and Y fluctuate about an equilibrium point as they act on each other.

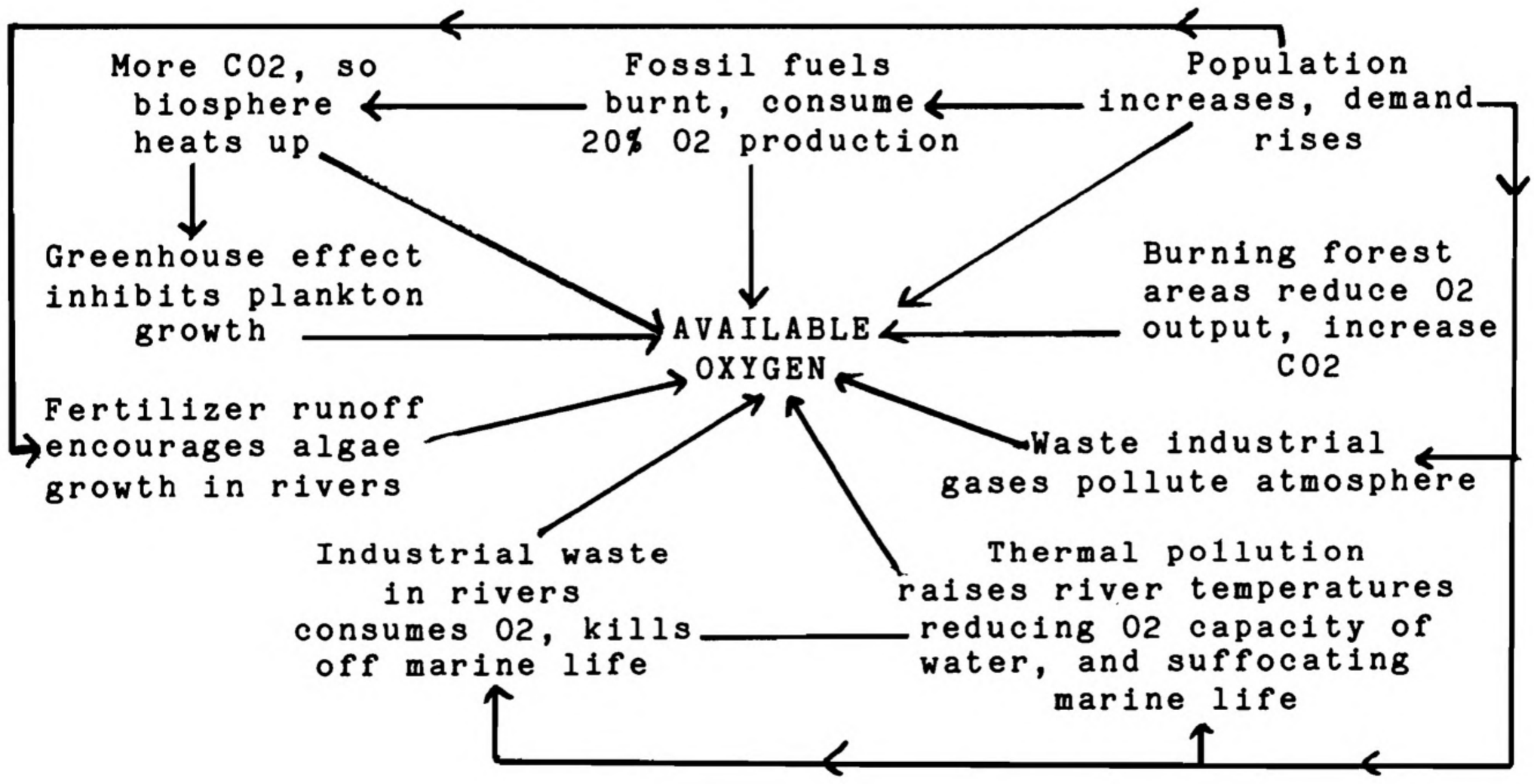
Your own simulation programs may well be easier to write if you consider in advance the behaviour of the system you want to model, in terms of positive or negative feedback loops, and the effect these will have on the system seeking an equilibrium point.

Some relationships are easier to see than others. For example, this diagram shows the relationship between the work involved in pumping water from a well, the thirst produced by that work, and the amount of water produced:



This is a fairly clear situation, and one which is probably simpler than any real-life situation you intend to model with a simulation system. If a double loop is needed to show the relationships of something simple like this, it is highly likely that the area you are studying will need two, or more, interacting loops.

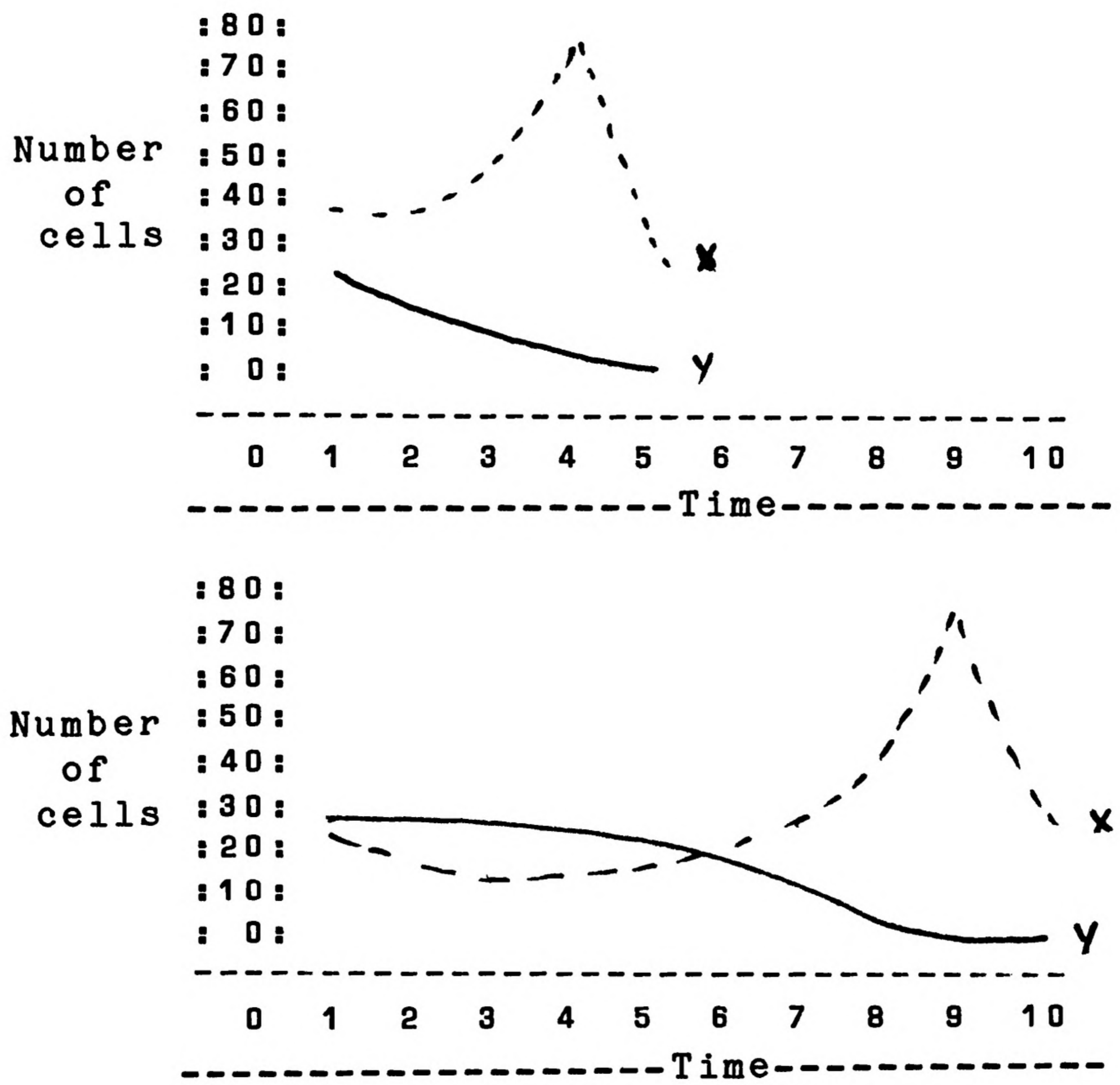
For example, oxygen use and replenishment on earth, can be shown with the following diagram:



Time Against Action

You may well find that producing a graph of the action of a system makes it easier to understand both the dynamics of the system, and the causal loops which are acting within it.

If we plot the populations of CELL X and CELL Y, against time, from the runs of chapter five's 'predator/prey' simulation, we get graphs like these:



These graphs show dramatically how the two life forms are related. As well, the crossover point of the lines suggests where the equilibrium points of the system may be.

So, the causal loop diagrams help us find the underlying relations of the system we hope to simulate, and a graph of the outputs of the simulation against time allow us to see cause and effect in action within the system. This graphed data can help us to see the effectiveness (or otherwise) of our simulation, and assists our understanding of what is actually going on as the simulation runs.

You now have enough background information to understand the processes involved in the creation and running of the simulation programs which make up the rest of the book. Once you get the programs up and running, you may well want to draw causal

diagrams to show the links between elements in these programs. As well, the output from some (but not all) of the simulations can be graphed, giving further insight into the working of the system being modelled.

Chapter Three
Feedback in Action

In this chapter, we'll look at our first simulation program, one which has been around – in one form or another – since the earliest days of commercial computers. A lot of employee's expensive computer time was consumed in those days as would-be space pilots tried to land creaky old space ships on the moon.

Unlike the predator/prey simulations which use simultaneous equations, simulations like this one use a number of formulae held within a loop. After each run through the loop, the system operator receives feedback on the status of particular variables which are being modified. On the basis of this feedback, the operator decides what input to enter, and then waits for the computer to report back to him or her on the result of processing that input within the loop.

Here is the series of mathematical operations within the loop (where A$ is the user input, signifying thrust):

```
160 THRUST=VAL[A$]+.1
170 FUEL=FUEL-THRUST
180 FLAG=THRUST-2
190 THRUST=0
200 HEIGHT=HEIGHT+VELOCITY+FLAG/4
210 VELOCITY=VELOCITY+FLAG
```

The loop has certain exit conditions (such as, in this simulation, when you crash, run out of fuel, or manage to land safely) which abort the cycle and end the run with a status report to the operator. Lines 22Ø and 24Ø in this next section of the program check to see if the cycle should leave the loop, while 23Ø sends it back for another run through:

```
220 IF HEIGHT<=10 THEN 240
230 IF HEIGHT>10 THEN 120
240 IF VELOCITY>-9 AND VELOCITY<5 THEN 2
```

Of course, you don't need to know this, or keep it in mind, when running a program like the *Space Landing Simulation* in this chapter. So far as you're concerned, you're trying to land your ship, and the mechanics of handling your input are none of your concern.

User input is, as I said a short while ago, limited to thrust. you signify a high thrust by pressing the "9" key; a low thrust with the "1"; and zero thrust with the "Ø". Intermediate values are entered by touching other number keys. There is no need to press the RETURN (or ENTER) key on this, as INKEY$ (or GET$) are used to read the keyboard.

Let's see how an unskilled pilot makes out when trying to land a space craft using this simulation:

```
FUEL          VELOCITY      HEIGHT
208.33        -11.42        497.03
208.23        -9.520001     507.99
208.13        -7.62         517.04
208.03        -5.72         524.2
207.93        -3.82         529.45
207.83        -1.92         532.8
207.73        -.02          534.26
207.63         1.88         533.81
207.53         3.78         531.47
207.43         5.68         527.22
207.33         7.58         521.07

188.23         10.48        341.57
179.13         3.38         332.87
170.03        -3.72         331.28
160.93        -10.82        336.78
151.83        -17.92        349.39
151.73        -16.02        366.84
151.63        -14.12        382.39
151.53        -12.22        396.05
151.43        -10.32        407.8
151.33        -8.42         417.66
151.23        -6.52         425.61
```

```
150.03          16.28           372.86
149.93          18.18           356.11
149.83          20.08           337.47
149.73          21.98           316.92
149.63          23.88           294.48
149.53          25.78           270.13
149.43          27.68           243.88
149.33          29.58           215.74
 149.23         31.48           185.69
 149.13         33.38           153.75
 149.03         35.28           119.9
 148.93         37.18           84.15
 148.83         39.08           46.51
--------------------------------
YOU HAVE CRASHED INTO THE SURFACE...
--------------------------------
FINAL INSTRUMENT READINGS WERE:
 FUEL           VELOCITY        HEIGHT
 148.73         40.98
--------------------------------
NEW CRATER ON MOON 2.25 METERS DEEP!
YOUR SKILL RATING IS -438
```

The picture is very different when a skilled space jockey takes the controls:

```
FUEL            VELOCITY        HEIGHT
210.71          4.02            496.05
210.61          5.92            491.57
210.51          7.82            485.18
210.41          9.72            476.89
210.31          11.62           466.7
210.21          13.52           454.61
210.11          15.42           440.62
210.01          17.32           424.73
209.91          19.22           406.94
200.81          12.12           389.5
191.71          5.02            379.16
182.61         -2.08            375.92
182.51         -.18             377.53
```

```
182.41          1.72            377.24
182.31          3.62            375.05
182.21          5.52            370.96
182.11          7.42            364.97
182.01          9.32            357.08

150.91          20.22           126.75
147.81          19.12           106.81
144.71          18.02           87.97
140.61          15.92           70.48
134.51          11.82           55.59
126.41          5.72            45.3
124.31          5.62            39.61
122.21          5.52            34.02
118.11          3.42            29.03
116.01          3.32            25.64
113.91          3.22            22.35
111.81          3.12            19.16
108.71          2.02            16.32
107.61          2.92            14.08
106.51          3.82            10.94
YOU HAVE LANDED SAFELY!
YOUR SKILL RATING IS 9067
-------------------------------
FINAL INSTRUMENT READINGS WERE:
 FUEL           VELOCITY        HEIGHT
 105.41         4.72            0
-------------------------------
```

Here's the listing so you can do your bit for NASA:

```
10 REM SPACE LANDING SIMULATION I
20 RANDOMIZE VAL(RIGHT$(TIME$,2))
30 REM ******************************
40 REM SET STARTING VALUES
50 FUEL=200+RND(1)*40
60 VELOCITY=RND(1)*20-6
70 HEIGHT=500-RND(1)*10
80 CLS
90 PRINT " FUEL";TAB(12);" VELOCITY";TAB
(24);" HEIGHT"
100 REM ******************************
```

```
110 REM MAJOR CYCLE
120 GOSUB 430
130 IF FUEL<=0 THEN FUEL=0:THRUST=0:GOTO
 170
140 A$=INKEY$
150 IF A$<"0" OR A$>"9" THEN 140
160 THRUST=VAL(A$)+.1
170 FUEL=FUEL-THRUST
180 FLAG=THRUST-2
190 THRUST=0
200 HEIGHT=HEIGHT+VELOCITY+FLAG/4
210 VELOCITY=VELOCITY+FLAG
220 IF HEIGHT<=10 THEN 240
230 IF HEIGHT>10 THEN 120
240 IF VELOCITY>-9 AND VELOCITY<5 THEN 2
90
250 GOSUB 410
260 PRINT "YOU HAVE CRASHED INTO THE SUR
FACE..."
270 IF HEIGHT>0 THEN HEIGHT=-HEIGHT
280 GOTO 320
290 PRINT "YOU HAVE LANDED SAFELY!"
300 PRINT "YOUR SKILL RATING IS"INT(-100
0*FUEL/(VELOCITY-HEIGHT))
310 HEIGHT=0
320 GOSUB 410
330 PRINT "FINAL INSTRUMENT READINGS WER
E:"
340 PRINT " FUEL";TAB(12);" VELOCITY";TA
B(24);" HEIGHT"
350 GOSUB 430
360 GOSUB 410
370 IF HEIGHT>=0 THEN END
380 IF HEIGHT<0 THEN PRINT "NEW CRATER O
N MOON"INT(ABS(100*(HEIGHT+.2)/3))/100"M
ETERS DEEP!"
390 PRINT "YOUR SKILL RATING IS "INT(100
*FUEL/(VELOCITY-HEIGHT))
400 END
410 PRINT "-----------------------------
---"
```

```
420 RETURN
430 PRINT INT(100*FUEL)/100;
440 PRINT TAB(12);-INT(100*VELOCITY)/100
;
450 IF HEIGHT>=0 THEN PRINT TAB(24);INT(
100*HEIGHT)/100
460 IF HEIGHT<0 THEN PRINT
470 RETURN
```

Once you can land consistently with the program in its current form, change line 5Ø to the following, to create a whole new challenge:

```
50 FUEL=100+RND(1)*40
```

Chapter Four
Monte Carlo Simulations

The term 'Monte Carlo Simulations' is given to those simulations which show the results of chance factors. The random elements in such programs may be weighted to simulate probabilities within specified limits (such as the results of throwing a pair of dice) or may be more or less genuinely random. In this chapter, you'll see a fairly 'open' Monte Carlo simulation in action.

Our program is designed to simulate the effect of Brownian movement on a very small particle suspended in a fluid. The random motion of such a particle (which is observed when the particle is less than about one-thousandth of a millimetre in diameter) is caused by the impact of the atoms or molecules of the fluid on the particle. Brownian movement can be seen as smoke disperses in still air, or ink is diffused in a tumbler of water.

Brownian movement is named after Robert Brown, a Scottish botanist who in 1827 first noted (but was unable to explain) the movement when looking through a microscope at a solution of pollen grains in water.

The simulation occurs on a 1Ø by 1Ø grid, with a single pollen grain, shown as "Ø". When you run the program, you enter the position where you want the grain to start, and as well you set up a 'goal' position. The simulation ends when the pollen reaches the goal position.

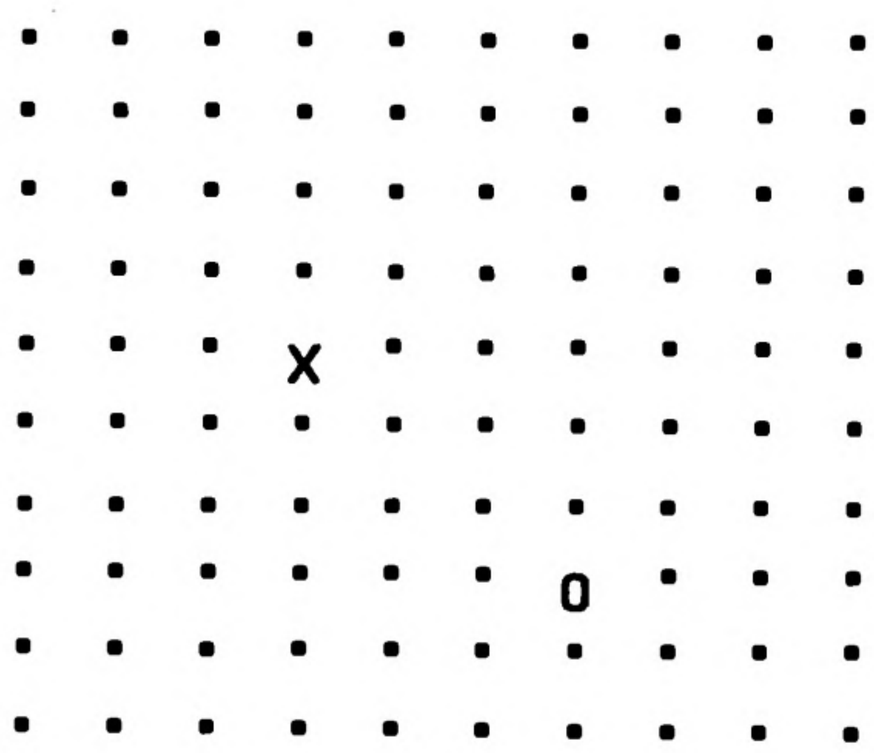
Here's how it begins:

```
ENTER FIRST START CO-ORDINATE [LESS
THAN 10]
? 3
```

```
ENTER SECOND START CO-ORDINATE (LESS
THAN 10)
? 4

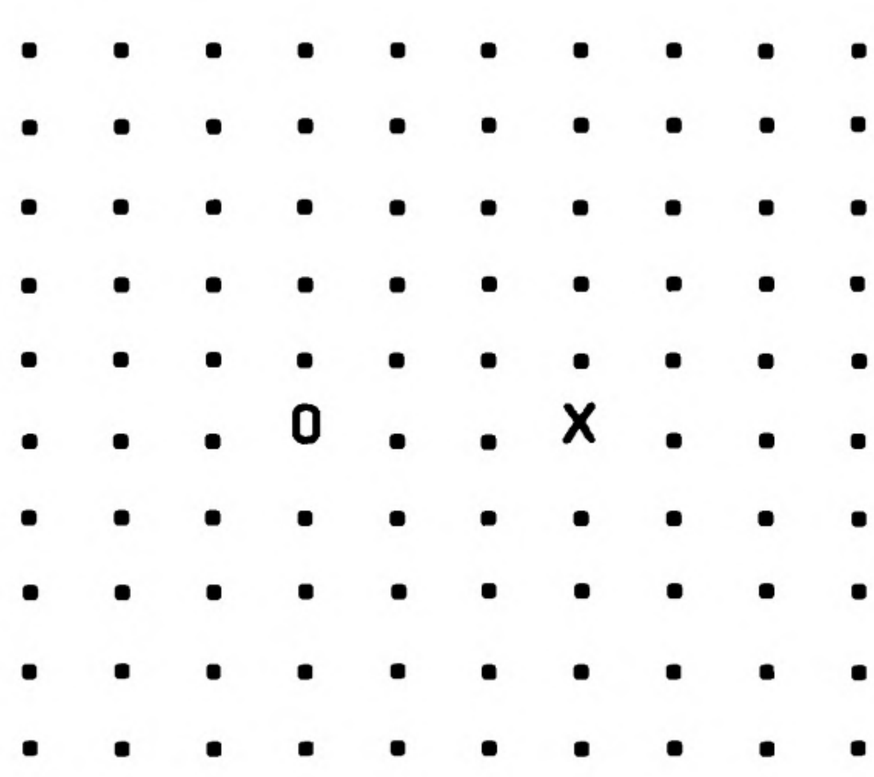
ENTER FIRST END CO-ORDINATE (LESS
THAN 10)
? 6
ENTER SECOND END CO-ORDINATE (LESS
THAN 10)
? 7

MOVE 1
```



And here are some screen printouts as the Brownian movement makes the grain drift about:

```
MOVE 5
```



```
MOVE 9
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . O X . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

```
MOVE 11
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . O . . .
. . . . . X . . .
. . . . . .
. . . . . .
. . . . . .
```

Finally, on the 14 move, the pollen grain reaches the target position, and the demonstration ends:

```
MOVE 14
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . O . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

```
DEMONSTRATION OVER
```

Here's the listing so you can explore this Monte Carlo demonstration:

```
10 REM MONTE CARLO DEMONSTRATION
20 GOSUB 370:REM INITIALIZE
30 REM ****************************
40 REM MAJOR CYCLE
50 GOSUB 100:REM PRINT
60 IF P=EP AND Q=EQ THEN PRINT:PRINT "DE
MONSTRATION OVER":END
70 GOSUB 230:REM GENERATE MOVE
```

```
80 GOTO 50
90 REM ****************************
100 REM PRINTOUT
110 A$(P,Q)="O"
120 M=M+1
130 CLS:PRINT:PRINT
140 PRINT "MOVE"M
150 FOR X=1 TO 10
160 FOR Y=1 TO 10
170 PRINT A$(X,Y);" ";
180 NEXT Y
190 PRINT
200 NEXT X
210 RETURN
220 REM ****************************
230 REM GENERATE MOVE
240 A$(P,Q)="."
250 G=0
260 T=INT(RND(1)*4)+1
270 ON T GOSUB 310,320,330,340
280 IF G=0 THEN 260
290 IF G=1 AND RND(1)>.5 THEN 260
300 RETURN
310 IF P>1 THEN P=P-1:G=G+1:RETURN
320 IF P<10 THEN P=P+1:G=G+1:RETURN
330 IF Q>1 THEN Q=Q-1:G=G+1:RETURN
340 IF Q<10 THEN Q=Q+1:G=G+1:RETURN
350 RETURN
360 REM ****************************
370 REM INITIALIZE
380 CLS
390 RANDOMIZE VAL(RIGHT$(TIME$,2))
400 DIM A$(10,10)
410 FOR X=1 TO 10
420 FOR Y=1 TO 10
430 A$(X,Y)="."
440 NEXT Y
450 NEXT X
460 PRINT:PRINT
470 PRINT "ENTER FIRST START CO-ORDINATE
 (LESS      THAN 10)"
```

```
480 INPUT P
490 IF P<1 OR P>10 THEN 480
500 PRINT "ENTER SECOND START CO-ORDINAT
E [LESS     THAN 10]"
510 INPUT Q
520 IF Q<1 OR Q>10 THEN 510
530 PRINT:PRINT
540 PRINT "ENTER FIRST END CO-ORDINATE [
LESS        THAN 10]"
550 INPUT EP
560 IF EP=P OR EP<1 OR EP>10 THEN 550
570 PRINT "ENTER SECOND END CO-ORDINATE
[LESS       THAN 10]"
580 INPUT EQ
590 IF EQ=Q OR EQ<1 OR EQ>10 THEN 580
600 A$(P,Q)="O"
610 A$(EP,EQ)="X"
620 RETURN
```

Chapter Five
Simultaneous Equations

We can create simulations which make use of simultaneous equations to show the effects of two interacting factors on each other. These equations are often used in 'predator/prey' simulations, such as the one we will examine in this chapter.

We have two forms of cell, living in a culture. Both cells feed on nutrients in the culture medium, and attempt to reproduce as the simulation runs. Cell X also needs to consume cell Y in order to survive. If there are too many of cell X, it will kill off all of cell Y, and will then die itself. If there are none of cell X, cell Y will reproduce wildly, and choke the medium.

You are allowed to set the starting numbers of cells X and Y in each run of the simulation. You have to try and create a population balance which will allow the cell colony to survive for as long as possible. Although the 'degree of aggressiveness' of cell X towards Y changes from run to run of the program (so you can't just learn two numbers which will always work), it does not change within a single run, so you can try various combinations to see how they behave.

The program first tells you the 'decay factor', which is the degree of aggressiveness cell X displays towards Y. It then asks you to enter the starting populations:

```
DECAY FACTOR IS .1408966

ENTER NUMBER OF CELL X TO
START [LESS THAN 40]
? 37
```

```
WE HAVE 37 X CELLS


ENTER NUMBER OF CELL Y TO
START (LESS THAN 40)
? 25
```

Once you've done this, the program gets underway, and works out what is going on, keeping you informed:

```
----------------
TIME ELAPSED: 1
 37 CELL X
 25 CELL Y
----------------
TIME ELAPSED: 2
 35 CELL X
 17 CELL Y
----------------
TIME ELAPSED: 3
 46 CELL X
 9 CELL Y
----------------
TIME ELAPSED: 4
 77 CELL X
 2 CELL Y
----------------
TIME ELAPSED: 5
 37 CELL X
 1 CELL Y


YOUR CELL CLASH SIMULATION SURVIVED
FOR 5 TIME PERIODS.
--------------------------------
THE BEST SURVIVAL TIME SO FAR IS 5
--------------------------------
DO YOU WANT A NEW RUN (Y OR N)?
```

The program also records the 'longest life' so far. It offers you subsequent runs so you can see the effect of different starting populations:

```
ENTER NUMBER OF CELL X TO
START [LESS THAN 40]
? 25

WE HAVE 25 X CELLS

ENTER NUMBER OF CELL Y TO
START [LESS THAN 40]
? 37

----------------
TIME ELAPSED: 4
 2 CELL X
 35 CELL Y
----------------
TIME ELAPSED: 5
 1 CELL X
 36 CELL Y

YOUR CELL CLASH SIMULATION SURVIVED
FOR 5 TIME PERIODS.
-----------------------------------.
THE BEST SURVIVAL TIME SO FAR IS


ENTER NUMBER OF CELL X TO
START [LESS THAN 40]
? 38

WE HAVE 38 X CELLS


ENTER NUMBER OF CELL Y TO
START [LESS THAN 40]
? 38
```

```
TIME ELAPSED: 2
 13 CELL X
 34 CELL Y
----------------
TIME ELAPSED: 3
 6 CELL X
 33 CELL Y
----------------
TIME ELAPSED: 4
 3 CELL X
 33 CELL Y
----------------
TIME ELAPSED: 5
 1 CELL X
 34 CELL Y


YOUR CELL CLASH SIMULATION SURVIVED
FOR 5 TIME PERIODS.
-----------------------------------
THE BEST SURVIVAL TIME SO FAR IS 5
-----------------------------------
DO YOU WANT A NEW RUN (Y OR N)?

ENTER NUMBER OF CELL X TO
START (LESS THAN 40)
? 22


WE HAVE 22 X CELLS


ENTER NUMBER OF CELL Y TO
START (LESS THAN 40)
? 30


----------------
TIME ELAPSED: 3
 13 CELL X
 23 CELL Y
```

```
----------------
TIME ELAPSED: 4
 14 CELL X
 21 CELL Y
----------------
TIME ELAPSED: 5
 15 CELL X
 18 CELL Y
----------------
TIME ELAPSED: 6
 19 CELL X
 15 CELL Y
----------------
TIME ELAPSED: 7
 27 CELL X
 12 CELL Y
----------------
TIME ELAPSED: 8
 41 CELL X
 7 CELL Y
----------------
TIME ELAPSED: 9
 74 CELL X
 2 CELL Y
----------------
TIME ELAPSED: 10
 32 CELL X
 1 CELL Y

YOUR CELL CLASH SIMULATION SURVIVED
FOR 10 TIME PERIODS.
--------------------------------
THE BEST SURVIVAL TIME SO FAR IS 10
--------------------------------
DO YOU WANT A NEW RUN [Y OR N]?
OK
```

The most crucial part of the program is the equations which do the work. These are in lines 33Ø and 34Ø:

```
330  CP=CP+[[8*CP-CP*EP/3]*FD]
340  EP=EP+[[4*EP-EP*CP]*.01]
```

Modify parts of them, and see what effect this has on the progress of your simulation. You'll probably be surprised to see how far-reaching are the effects of even the most apparently minor change. Here's the listing:

```
10 REM SIMULTANEOUS EQUATIONS
20 CLS
30 RANDOMIZE VAL[RIGHT$[TIME$,2]]
40 HS=0
50 FD=RND[0]
60 PRINT:PRINT "DECAY FACTOR IS"FD
70 GOSUB 550
80 CLS
90 PRINT:PRINT
100 PRINT "ENTER NUMBER OF CELL X TO
          START [LESS THAN 40]"
110 INPUT CP:IF CP<1 OR CP>39 THEN 110
120 PRINT:PRINT
130 PRINT "WE HAVE"CP"X CELLS"
140 PRINT:PRINT
150 PRINT "ENTER NUMBER OF CELL Y TO
          START [LESS THAN 40]"
160 INPUT EP:IF EP<1 OR EP>39 THEN 160
170 CLS:PRINT:PRINT "PLEASE STAND BY..."

180 GOSUB 550:CLS
190 DA=1
200 IF CP>EP/FD THEN CP=EP/FD
210 PRINT "----------------"
220 PRINT "TIME ELAPSED:"DA
230 PRINT INT[CP]"CELL X"
240 PRINT INT[EP]"CELL Y"
250 REM **************************
260 REM MAJOR CYCLE
270 GOSUB 550
280 DA=DA+1
290 PRINT "----------------"
300 PRINT "TIME ELAPSED:"DA
310 IF CP>EP/FD THEN CP=EP/FD
320 REM EQUATIONS FOLLOW; MODIFY PARTS
       OF THEM TO SEE WHAT HAPPENS
330 CP=CP+[[8*CP-CP*EP/3]*FD]
```

```
340 EP=EP+[[4*EP-EP*CP]*.01]
350 PRINT INT[CP]"CELL X"
360 PRINT INT[EP]"CELL Y"
370 IF EP<2 OR CP<2 THEN 410
380 GOSUB 550
390 GOTO 280
400 REM ****************************
410 IF DA>HS THEN HS=DA
420 PRINT:PRINT
430 PRINT "YOUR CELL CLASH SIMULATION SU
RVIVED"
440 PRINT "FOR"DA"TIME PERIODS."
450 PRINT "-------------------------
-----"
460 PRINT "THE BEST SURVIVAL TIME SO FAR
 IS"HS
470 GOSUB 550
480 PRINT "-------------------------
-----"
490 PRINT "DO YOU WANT A NEW RUN [Y OR N
]?"
500 A$=INKEY$
510 IF A$<>"Y" AND A$<>"N" THEN 500
520 IF A$="Y" THEN CLS:GOTO 60
530 PRINT "OK":PRINT:PRINT:END
540 REM ****************************
550 FOR J=1 TO 2000:NEXT J
560 RETURN
```

Chapter Six
The Meaning of Life

In the previous chapter, we looked at a program which simulated the action of two 'competing' life forms occupying a single space. In that simulation, the growth and decay of the cells were governed by simultaneous equations.

In this chapter, we'll look at two versions of one of the old simulation classics, John Conway's *Life*. This simulation concerns the growth and development of a colony of cells living on a grid.

Developed by Conway when he was at Gonville and Caius College, in Cambridge, England, *Life* produces some amazing effects, whose richness is not even hinted at by the rules under which the cells live, reproduce and die. To show this, we'll look at a colony of cells which starts off forming a square. This particular colony dies off after these three 'generations':

```
GENERATION 1                    GENERATION 2


                                      XXX             XXX
XXXXX           XXXXX               XXXXX           XXXXX
X   X           X   X              XX X XX         XX X XX
X   X           X   X              XXX XXX         XXX XXX
X   X           X   X              XX X XX         XX X XX
XXXXX           XXXXX                XXXXX           XXXXX
                                       XXX             XXX
```

```
GENERATION 3


      X                   X
    X   X               X   X
  X       X           X       X
X           X       X           X

  X       X           X       X
    X   X               X   X
      X                   X
```

Another colony, which begins life as an 'X', evolves as follows (the program prints a 'mirror image' colony to the right of the original colony in all these sample runs):

```
GENERATION 1                    GENERATION 2

X   X                           X   X
 X X                             X X
  X                               X
 X X                             X X
X   X                           X   X

GENERATION 3                    GENERATION 4

  X                               X
 X X                             XXX
X   X                           XX XX
 X X                             XXX
  X                               X
```

GENERATION 5

```
 XXX
X   X
X   X
X   X
 XXX
```

```
 XXX
X   X
X   X
X   X
 XXX
```

GENERATION 6

```
  X
 XXX
X X X
XXX XXX
X X X
 XXX
  X
```

```
  X
 XXX
X X X
XXX XXX
X X X
 XXX
  X
```

GENERATION 7

```
  XXX

X   X
X   X
X   X

  XXX
```

```
  XXX

X   X
X   X
X   X

  XXX
```

GENERATION 8

```
  X
  X
  X

XXX XXX
  X
  X
  X
```

```
  X
  X
  X

XXX XXX
  X
  X
  X
```

If we continue to run this simulation, each horizontal set of X's flips to become a vertical set, and the vertical sets become horizontal. This pattern of three which alternates from horizontal to vertical is one of a number of patterns which regularly crop up in runs of the program. This particular pattern is called 'traffic lights'.

The Rules

Conway's rules are not very complicated, but give rise to the marvellous effects we've been looking at. They assume each cell has eight neighbours (and we keep the outermost ring of cells empty; these are checked, but not printed out). If a cell has two or three neighbours, it survives to the next generation. If a cell has four neighbours, it dies out in the next generation, due to overcrowding. If there are exactly three cells neighbouring an empty cell, a cell is born in that location in the next generation.

The rules are applied simultaneously across the whole grid. We do this by having a second grid, which records the changes as the first grid is scanned. The second grid is then copied into the first, before it is printed.

This is the listing which produced the effects above. The DATA statements at the end can be, of course, changed to whatever initial colony you desire:

```
10 REM CONWAY'S LIFE SIMULATION
20 REM DEFINED INITIAL COLONY
30 GOSUB 460:REM INITIALIZE
40 REM ****************************
50 REM MAJOR CYCLE
60 GENERATION=GENERATION+1
70 GOSUB 290:REM PRINTOUT
80 GOSUB 110:REM EVOLVE
90 GOTO 60
100 REM ****************************
110 REM EVOLVE
120 FOR X=2 TO 12
130 FOR Y=2 TO 12
140 C=0
150 IF A$[X-1,Y-1]="X" THEN C=C+1
160 IF A$[X-1,Y]="X" THEN C=C+1
170 IF A$[X-1,Y+1]="X" THEN C=C+1
180 IF A$[X,Y-1]="X" THEN C=C+1
190 IF A$[X,Y+1]="X" THEN C=C+1
200 IF A$[X+1,Y-1]="X" THEN C=C+1
210 IF A$[X+1,Y]="X" THEN C=C+1
220 IF A$[X+1,Y+1]="X" THEN C=C+1
230 IF A$[X,Y]="X" AND C<>2 AND C<>3 THE
N B$[X,Y]=" "
240 IF A$[X,Y]=" " AND C=3 THEN B$[X,Y]=
"X"
250 NEXT Y
260 NEXT X
270 RETURN
280 REM ****************************
290 REM PRINTOUT
300 CLS
```

```
310 PRINT
320 PRINT TAB(4);"GENERATION"GENERATION
330 PRINT
340 FOR X=2 TO 12
350 FOR Y=2 TO 12
360 A$(X,Y)=B$(X,Y)
370 PRINT A$(X,Y);
380 NEXT Y
390 FOR Y=12 TO 2 STEP-1
400 PRINT A$(X,Y);
410 NEXT Y
420 PRINT
430 NEXT X
440 RETURN
450 REM **************************
460 REM INITIALIZATION
470 CLS
480 RANDOMIZE VAL(RIGHT$(TIME$,2))
490 DEFINT A-Z
500 DIM A$(13,13),B$(13,13)
510 PRINT:PRINT "   PLEASE STAND BY..."
520 FOR X=1 TO 13
530 PRINT 14-X;
540 FOR Y=1 TO 13
550 REM FILL ARRAY WITH BLANKS
560 A$(X,Y)=" "
570 B$(X,Y)=A$(X,Y)
580 NEXT Y
590 NEXT X
600 READ D:IF D=99 THEN 630
610 READ E:A$(D,E)="X":B$(D,E)="X"
620 GOTO 600
630 GENERATION=0
640 RETURN
650 DATA 5,5,5,9,6,6,6,8
660 DATA 7,7
670 DATA 8,6,8,8,9,5,9,9
700 DATA 99
```

Colonies at Random

If you prefer to let your computer generate a starting colony at random, leaving you simply to admire its work, add the following lines:

```
555 IF RND(1)>.5 THEN A$(X,Y)="X":GOTO 5
70
595 GOTO 630
```

The evolution of a randomly-generated colony is always fascinating to watch, as these sample runs indicate:

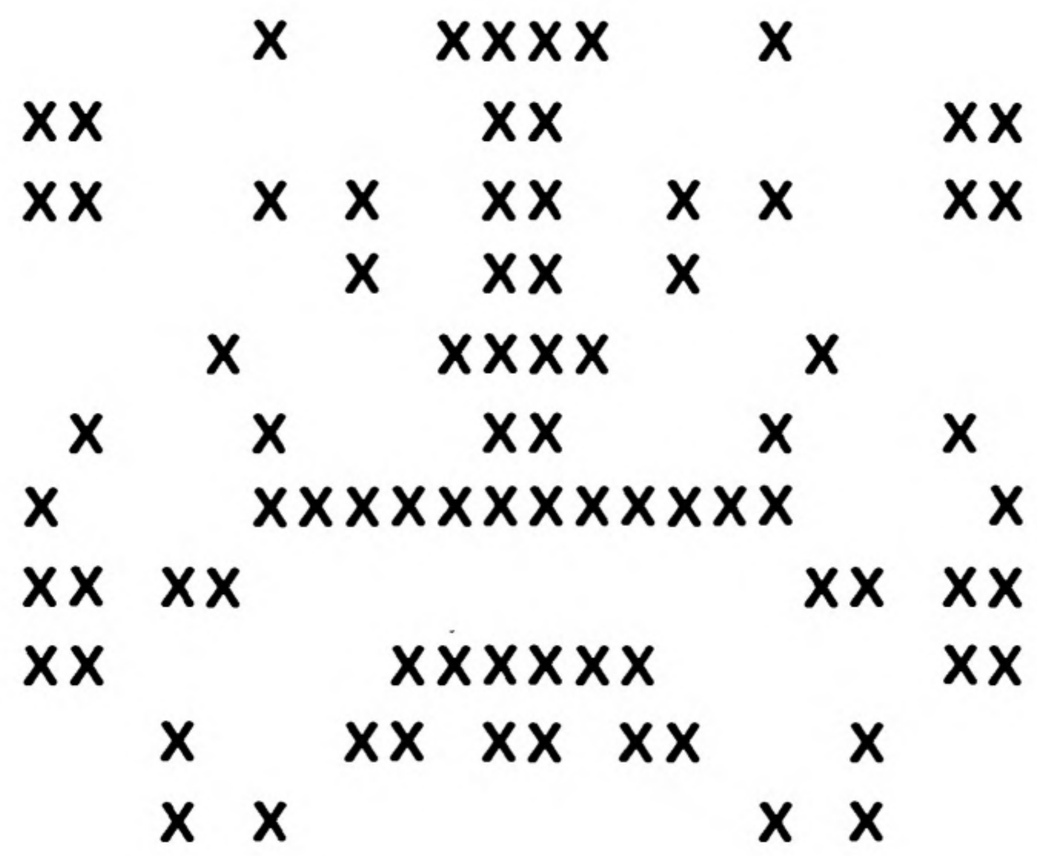
```
                GENERATION 1

  X   X XXXXXXXXXX X   X
X       XX  XX  XX       X
X  X X     XX     X X   X
 X       XXXXXX        X
    X      XX      X
    X   X          X   X
X XX      XXXXXX      XX X
 X    XXX X  X XXX    X
 X       XXXXXX       X
XXX                  XXX
 XXXXXX   XX   XXXXXX

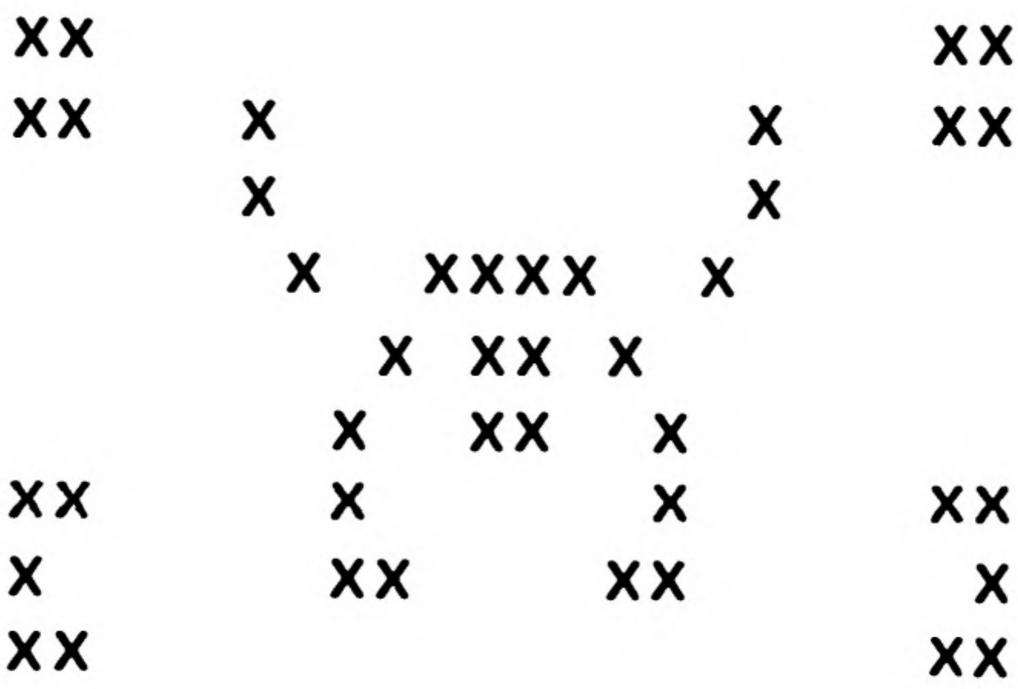
               GENERATION 2

      XX XXXXXX XX
XX   X      XX      X   XX
XX     XXX XX XXX     XX
     X       XX       X
             XX
   XXX       XX       XXX
  XXXXX  XXXXXX  XXXXX
XX     XX       XX     XX
       XXXXXXXXXXX
X    XX      XX      XX    X
X   XXX              XXX   X
```

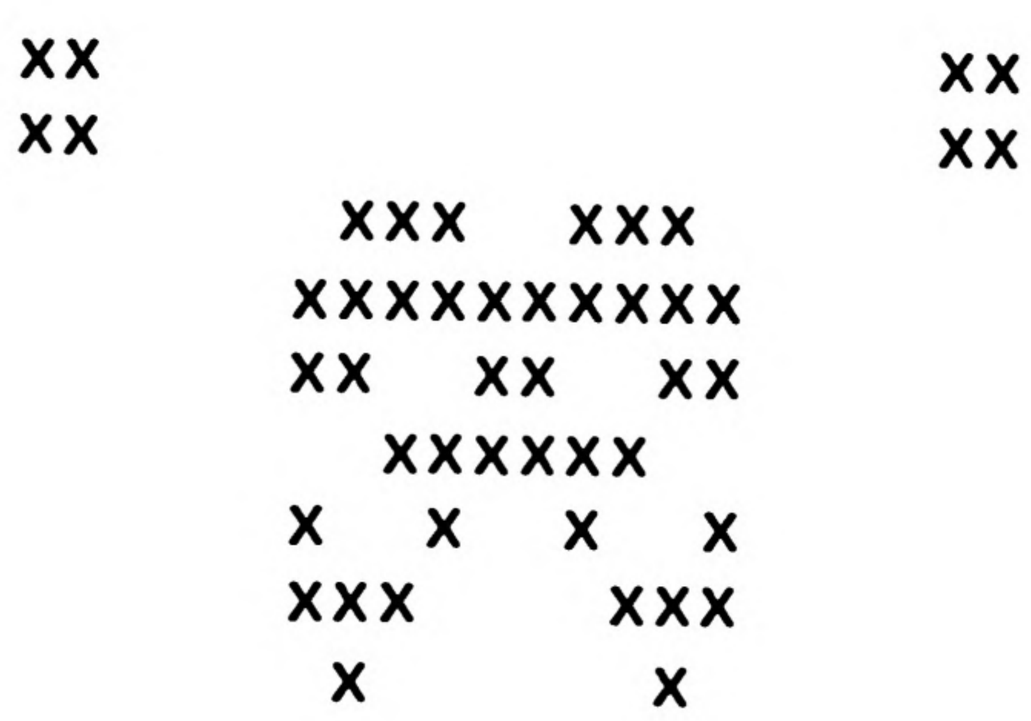
GENERATION 3



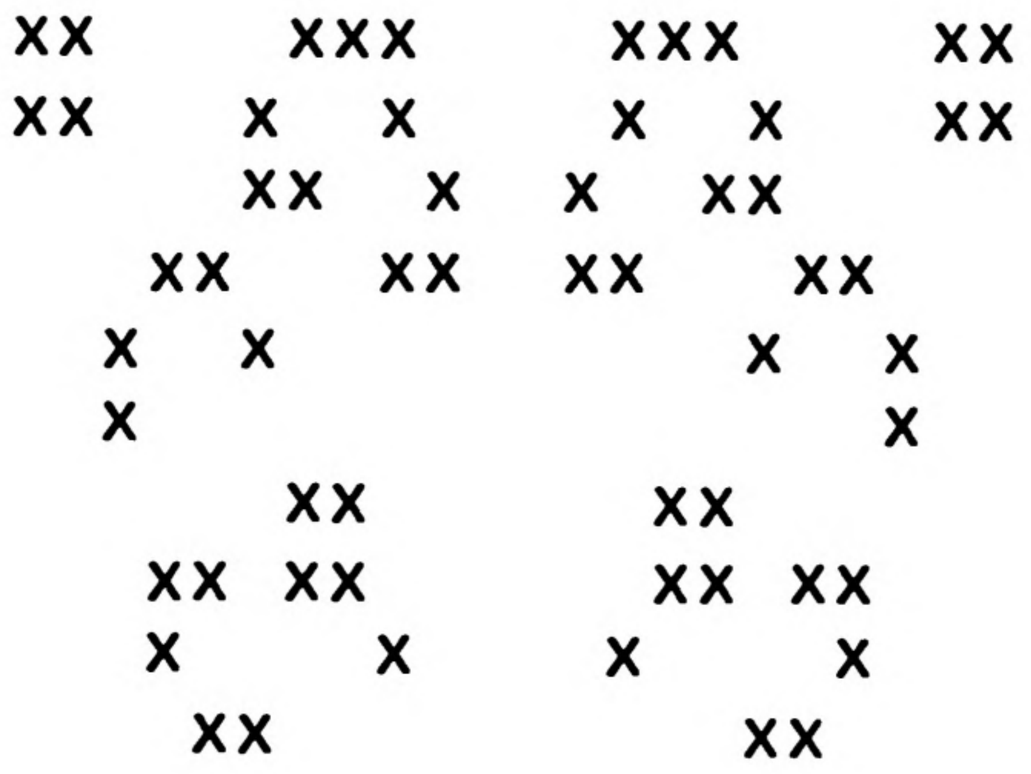
GENERATION 9



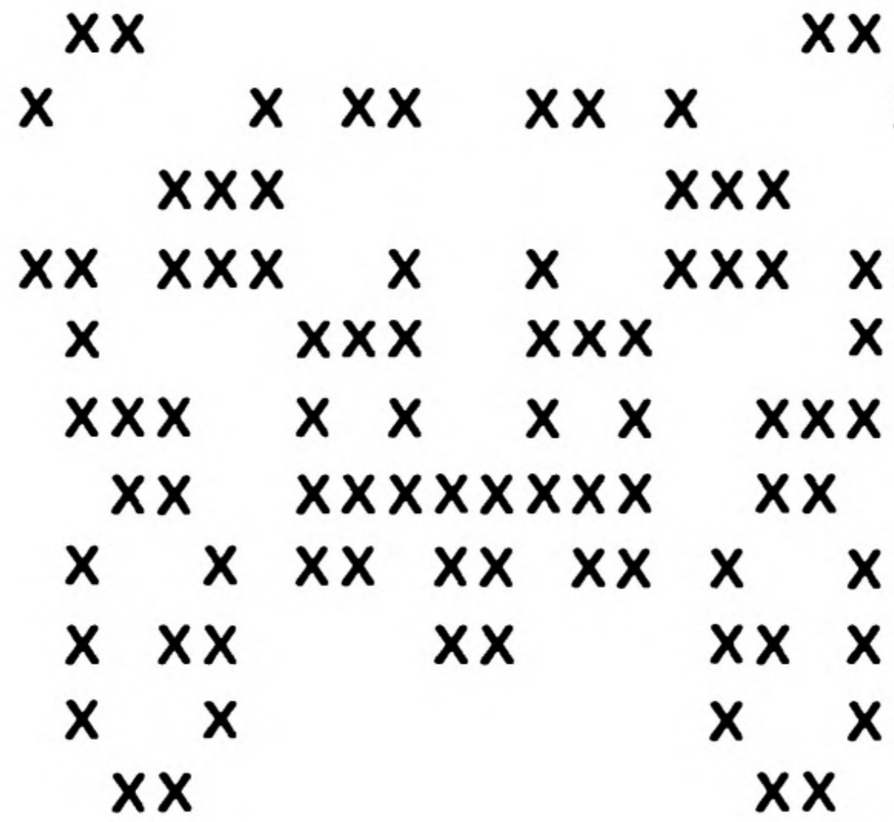
GENERATION 12



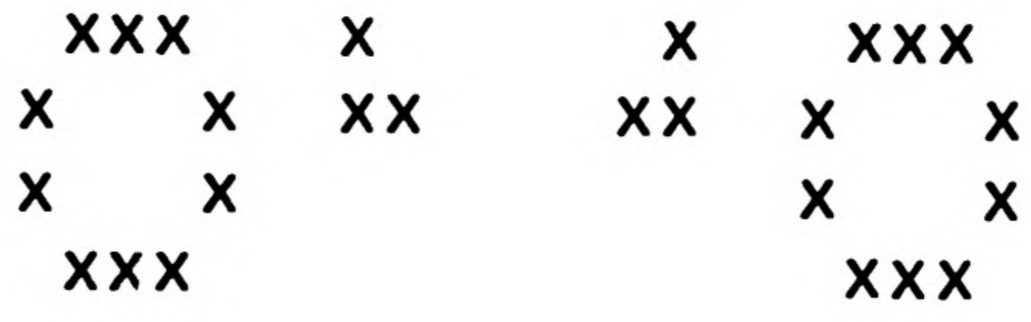
GENERATION 24



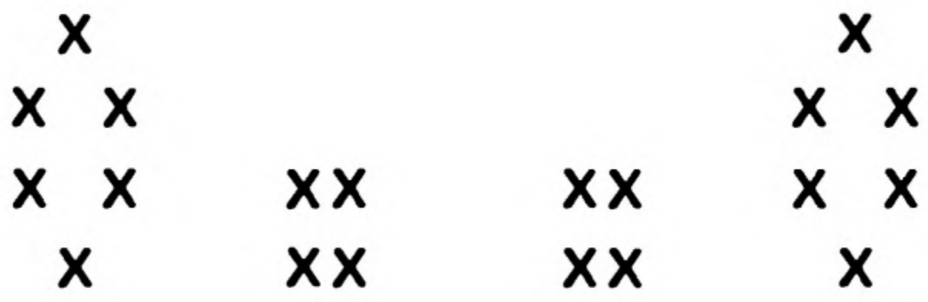
GENERATION 39



GENERATION 51



GENERATION 62



Chapter Seven
Robot Simulations

In this section of the book, we will create programs which perform two simulations at once. The programs will firstly allow us to simulate a robot who will follow a series of commands entered by us before the program is run. Secondly, the programs emulate computer language interpreters, which allow the computer (and hence the robot) to understand the instructions we have given in a new 'robot control language' (RCL) developed especially for this book.

Muse Software, of Baltimore, released a superb robot simulation program in mid-1983 under the name of *Robotwar*. This program allows you to pre-program a number of 'robots', before setting them loose on a VDU battlefield to fight it out for victory. *Robotwar* allows you to program the robots in a language which is essentially English, with a few BASIC commands (such as GOTO and GOSUB).

Our programs, *Robot Logo* and *Point-Duty Robot* (which is derived from the first program), are much simpler than *Robotwar,* yet still produce satisfying, and surprisingly interesting results.

We'll look first at *Robot Logo.* To show you what it can do, and to give you an idea of what an RCL listing looks like, here's a program which gets the robot to trace out my initials on the screen, leaving a trail as it goes:

```
GO 6,10
FACE 90
FORWARD 7
GO 8,26
FORWARD 5
PR
GO 6,16
FACE 180
FORWARD 6
GO 6,26
FORWARD 6
GO 6,34
FORWARD 6
PRINTOUT
*
```

This is the result of running that program:

```
X X X X X X X    X          X
      X          X          X
X              X X X X    X
X              X          X
X              X          X
X              X          X
```

There are 12 commands in RCL, any of which can be included in a program abbreviated to its first two letters. The RCL program is written into the overall BASIC program, as a set of up to 20 data statements, terminated with a data statement which just contains an asterisk. These data lines are near the beginning of the program. The current line being processed appears at the top of the screen as the program is running, so you can see the relevant line, and its effect, as it occurs.

The Language

Here is the complete vocabulary of RCL:
START (abbreviated to ST) – start again
PRINTOUT (PR) – stop the robot action, clear the screen, and print out the current situation, before continuing with the program run

FORWARD (FO) – this is followed by a single number, and it tells the robot to move forward the number of 'steps' indicated by the number which follows the command
BACK (BA) – the opposite, naturally enough, of FORWARD
TURN (TU) – followed by a number, this command turns the robot through the specified number of degrees; it starts a run at zero degrees, designated as the top of the screen
HOME (HO) – returns the robot to the center of the screen, facing upwards
CLEAN (CL) – cleans the previous steps away (the action of this command, like the others, will become clear once you use it)
GO – this is followed by two numbers, which are the coordinates to which the robot moves, without leaving a trail between its old position and the new one
RANDOM (RA) – moves the robot to a randomly-chosen position on the screen; acts as a random GO
REPEAT (RE) – followed by a number, this command allows you to cycle through a section of the program a specified number of times before continuing; this can produce some fascinating effects as you'll see shortly
END REPEAT (EN) – terminates the REPEAT cycle; all the commands in the RCL program which lie between REPEAT and END REPEAT will be cycled through the number of times specified
FACE (FA) – followed by a number, this turns the robot to face an absolute angle, with zero degrees at the top of the screen (whereas TURN is relative to the current angle the robot is facing)

The commands can be combined to produce many startling results. If you are at all familiar with Logo, you will recognise that several of the RCL commands are very similar to the turtle graphic commands in logo.

Logo was developed under Seymour Papert, with Marvin Minsky, Harold Abelson and Andrew diSessa, at the Massachusetts Institute of Technology by the Artificial Intelligence Group. It was written so that its structure would be closer to the way human beings solve problems than being heavily influenced by how machines run programs.

Of course, our RCL is pretty limited in comparison with Logo, although it can easily be extended to whatever level of complexity

you like. Studying the listing can show you how it is relatively easy to write a program which *interprets* another program written in a language which you have created.

Despite its limitations, you can have a lot of fun with RCL as the following sample runs indicate.

Octagons are produced by this RCL program:

```
OCTAGON

GO 11,35
REPEAT 8
FORWARD 4
TURN 45
END REPEAT
PRINTOUT
*
```

```
          X X X X X
        X         X
      X             X
    X                 X
  X                     X
  X                     X
  X                     X
  X                     X
    X                 X
      X             X
        X         X
          X X X X X
```

You'll get diamonds with this one:

```
DIAMONDS

RANDOM
FACE 45
REPEAT 4
FORWARD 2
TURN 90
END REPEAT
PRINTOUT
START AGAIN
*
```

```
            X
          X   X
        X       X
      X           X
        X       X
          X   X
            X
```

And squares are created by this RCL listing:

```
SQUARE
GO 11,35
REPEAT 4
FORWARD 6
TURN 90
END REPEAT
PRINTOUT
*
```

```
X X X X X X X
X           X
X           X
X           X
X           X
X           X
X X X X X X X
```

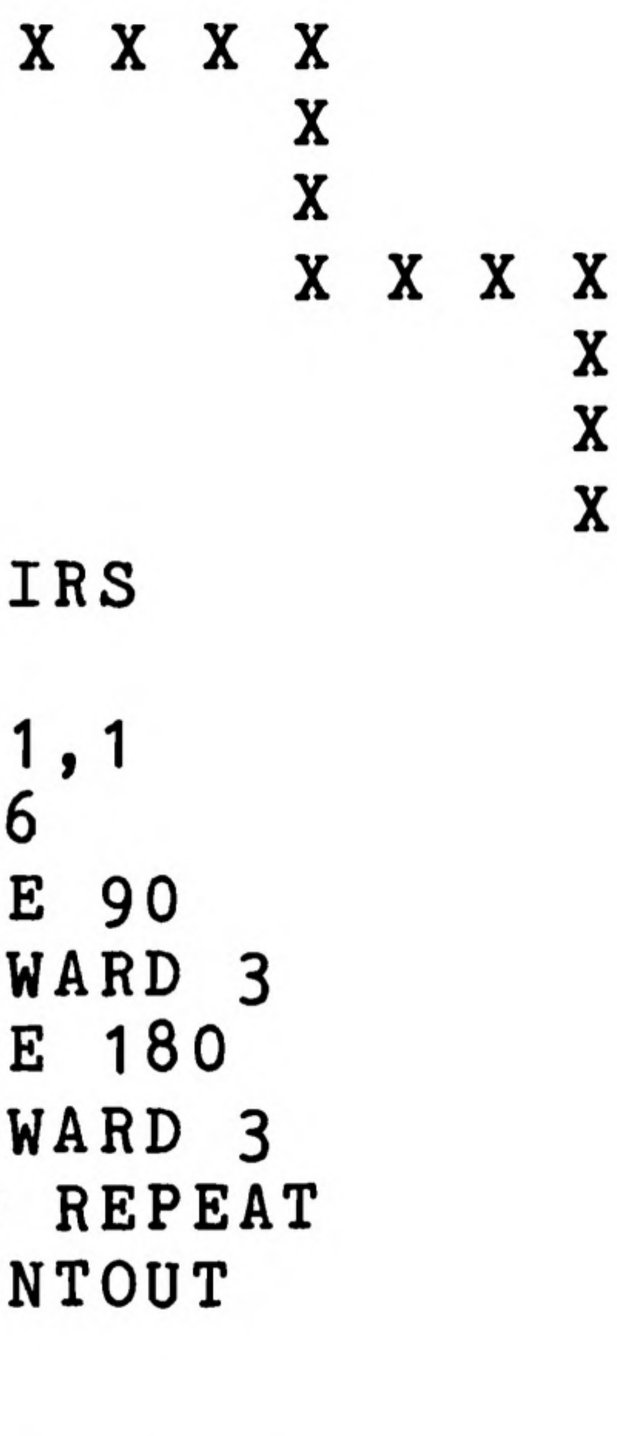
Descending triangles are easy (and note we're using abbreviated commands in this RCL listing):

```
DESCENDING
TRIANGLES

GO 5,5
RE 6
FO 3
TU 135
FO 3
TU 135
FO 3
FA 90
FO 3
FA 180
FO 3
FA 0
PR
EN
*
```

```
X
X X
X   X
X X X X
      X X
      X   X
      X X X X
            X X
            X   X
            X X X X
                  X X
                  X   X
                  X X X X
                        X
                        X
```

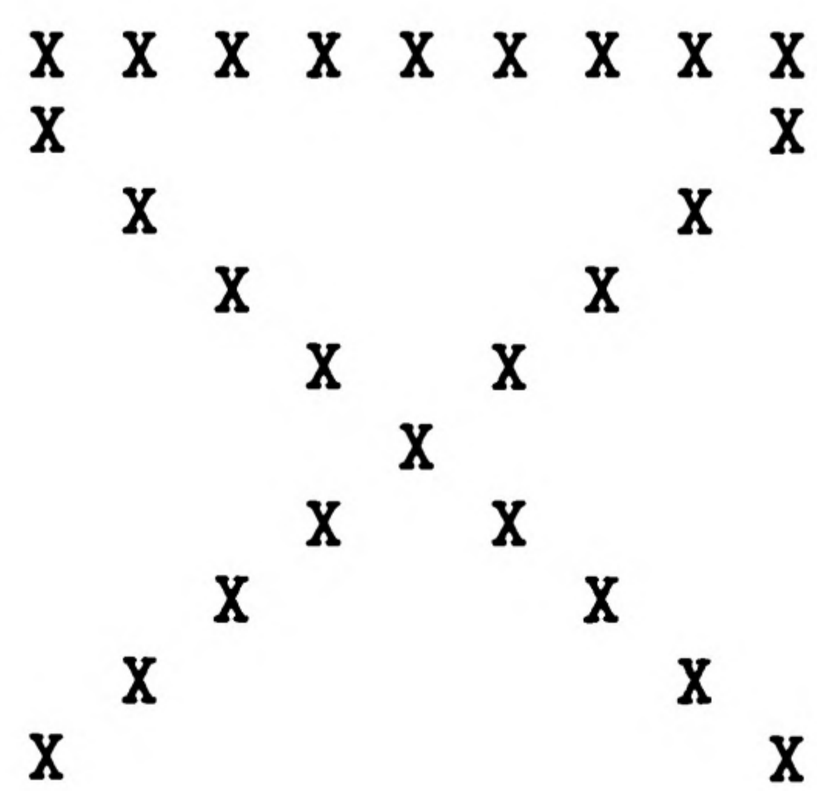
And from the next RCL program we can get our robot to walk down stairs:



```
STAIRS

GO 1,1
RE 6
FACE 90
FORWARD 3
FACE 180
FORWARD 3
END REPEAT
PRINTOUT
*
```

If you're feeling very creative, you can also get the robot to trace out pictures, like my initials earlier, or this 'side view of a table':



```
SIDE VIEW OF TABLE

GO 20,50
FACE 315
FORWARD 9
FACE 90
FORWARD 10
FACE 225
FORWARD 10
PRINTOUT
*
```

Writing a Program

As I said earlier, the RCL program is held in a series of DATA statements near the beginning of the listing:

```
100 REM ROBOT LOGO
110 GOSUB 1730:REM INITIALIZE
120 GOTO 490
130 REM ************************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ************************************
```

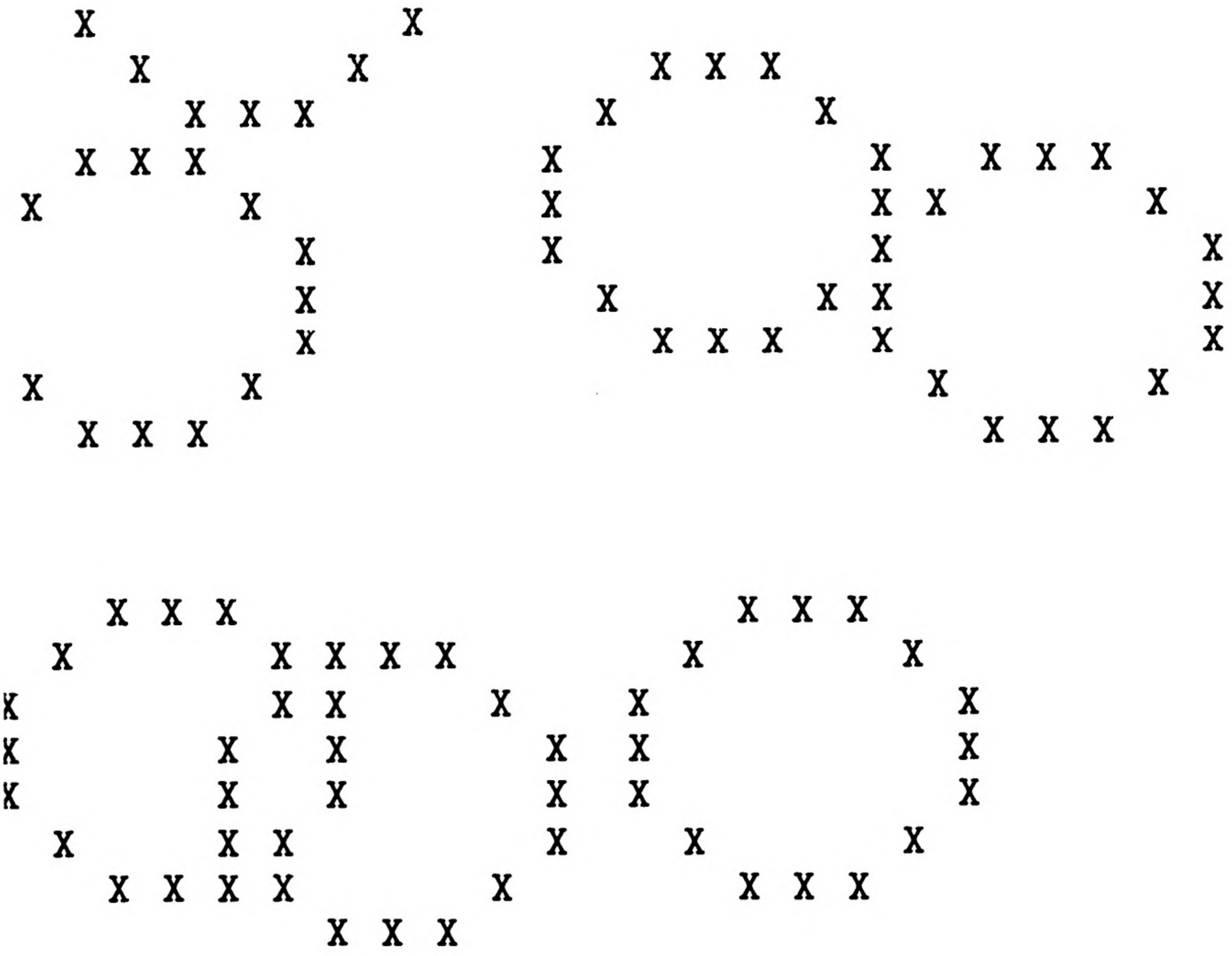
Here's a listing to create random octagons. We can learn a lot about RCL by examining it:

```
OCTAGON

RANDOM
REPEAT 8
FORWARD 2
TURN 45
END REPEAT
PRINTOUT
START AGAIN
*
```

It begins with RANDOM which moves the robot to a randomly-chosen position on your screen. The dimensions of the screen are included within the program (see lines 1770 and 1780 in the listing in the next chapter), so it will work easily on your system no matter what its size.

Once the robot is in place, the computer comes to the line REPEAT 8. This tells it, naturally enough, that the next section of the program (down to the REPEAT) is to be run through eight times. The robot moves FORWARD 2, then TURNs through 45 degrees. Once it has done this eight times, it has traced out an octagon which PRINTOUT puts on the screen for you to see:



The final command, START AGAIN, sends action back to the first line of the program, so select a new robot position at random. RCL automatically rejects values which are off the screen, so the program will not crash if you try to walk off the edge of the robot world. It will simply draw the required material, 'off the screen', as it were, and then continue at the correct space when it returns to the screen.

You can see, from the examples we've looked at, that the robot can be made to do a great deal. The biggest limitation is in the TURN and FACE commands. It can only work, in its present form, in increments of 45 degrees, and will change (see lines 1040 and 1620) any angle into the closest multiple of 45 before it sends the robot out along that path. This restriction was necessary in order to ensure that, despite the very rough resolution of the grid upon which the robot walks, a reasonable picture was still drawn by the robot's trail.

The full *Robot Logo* listing is in the next chapter.

Chapter Eight
The Robot Logo Listing

You're sure to have a lot of fun using this listing in its present form to get the robot to follow programs which you have devised. Simply place them, in the form of DATA "REPEAT 8", and the like, in lines 14Ø through to 31Ø. Line 32Ø, where you can just see an asterisk in the DATA statement, tells the program interpreter that it has come to the end of your listing. I got my program listings to the printer by running the program, then entering and running the following directly:

```
FOR Z=1 TO 20:LPRINT A$(Z):NEXT Z
```

In its present form the program can cope with RCL programs which are up to 2Ø lines long. If you want it to accept longer programs, simply change the 2Ø in lines 53Ø and 182Ø to the number of commands you want to include. Remember that you need to allow for a command slot to hold the asterisk (end of program flag) line.

Here, then, is the listing so you can run your own robot:

```
100 REM ROBOT LOGO
110 GOSUB 1730:REM INITIALIZE
120 GOTO 490
130 REM ********************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
```

```
220  REM
230  REM
240  REM
250  REM
260  REM
270  REM
280  REM
290  REM
300  REM
310  REM
320  DATA "*"
330  REM ****************************
340  REM INT UX,AX
350  UX=INT(UX+.5):AX=INT(AX+.5)
360  RETURN
370  REM ****************************
380  REM              PRINT OUT
390  CLS:REM OR LOCATE 1,1 OR SIMILAR
400  PRINT "STEP"PSN" > ";A$(PSN):PRINT
410  FOR J=1 TO DEPTH
420  FOR K=1 TO BREADTH
430  PRINT Z$(J,K);
440  NEXT K
450  PRINT
460  NEXT J
470  RETURN
480  REM ****************************
490  REM READ PROGRAM
500  COUNT=COUNT+1
510  READ A$(COUNT)
520  IF A$(COUNT)="*" THEN 550
530  IF COUNT<20 THEN 500
540  REM ****************************
550  REM EXECUTE PROGRAM
560  PSN=0:REM PROGRAM STEP NUMBER
570  PSN=PSN+1
580  IF PSN=21 THEN 580:REM END
590  FLAG=0
600  M$=A$(PSN)
610  IF M$="*" THEN 610:REM END
620  N$=LEFT$(M$,2)
```

```
630  IF  N$="ST"  THEN  560:REM  START  AGAIN
640  IF  N$="PR"  THEN  GOSUB  380:REM  PRINTO
UT
650  IF  N$="FO"  THEN  FLAG=1
660  IF  N$="BA"  THEN  FLAG=2
670  IF  N$="TU"  THEN  FLAG=3
680  IF  N$="HO"  THEN  FLAG=4
690  IF  N$="CL"  THEN  FLAG=5
700  IF  N$="GO"  THEN  FLAG=6
710  IF  N$="RA"  THEN  FLAG=7
720  IF  N$="RE"  THEN  FLAG=8
730  IF  N$="EN"  THEN  FLAG=9
740  IF  N$="FA"  THEN  FLAG=10
750  ON  FLAG  GOSUB  780,940,1000,1160,1220
,1260,1400,1460,1530,1580
760  GOTO  570
770  REM  **************************
780  REM                FORWARD
790  M$=MID$[M$,4]
800  IF  ASC[M$]=87  THEN  M$=MID$[M$,6]
810  F$="F"
820  NUM=VAL[M$]
830  FOR  E=1  TO  NUM
840  IF  UX<1  OR  UX>DEPTH  THEN  880
850  IF  AX<1  OR  AX>BREADTH  THEN  880
860  Z$[UX,AX]=T$
870  REM  DELETE  THE  '*2'  AT  END  OF  NEXT
 TWO  LINES  IF  BETTER  ON  YOUR  SYSTEM
880  IF  F$="F"  THEN  UX=UX+UP:AX=AX+AC*2
890  IF  F$="B"  THEN  UX=UX-UP:AX=AX-AC*2
900  GOSUB  340
910  NEXT  E
920  RETURN
930  REM  **************************
940  REM                BACK
950  M$=MID$[M$,4]
960  IF  ASC[M$]=75  THEN  M$=MID$[M$,3]
970  F$="B"
980  GOTO  820
990  REM  **************************
1000  REM                TURN
```

```
1010 M$=MID$(M$,4)
1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
1030 NUM=VAL(M$)
1040 Y=INT((NUM+17.5)/45)
1050 IF Y=0 OR Y=8 THEN RETURN
1060 FOR J=1 TO Y
1070 IF UP=-1 AND AC=0 THEN AC=1:GOTO 11
30
1080 IF UP=0 AND AC=1 THEN UP=1:GOTO 113
0
1090 IF UP=1 AND AC=0 THEN AC=-1:GOTO 11
30
1100 IF UP=0 AND AC=-1 THEN UP=-1:GOTO 1
130
1110 IF UP=-1 AND AC=-1 OR UP=1 AND AC=1
 THEN AC=0:GOTO 1130
1120 IF UP=-1 AND AC=1 OR UP=1 AND AC=-1
 THEN UP=0
1130 NEXT J
1140 RETURN
1150 REM ************************
1160 REM              HOME
1170 AX=INT((BREADTH+.5)/2)
1180 UX=INT((DEPTH+.5)/2)
1190 UP=-1:AC=0:REM FACES UP
1200 RETURN
1210 REM ************************
1220 REM              CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ************************
1260 REM           GO X,Y
1270 P=0
1280 P=P+1
1290 IF MID$(M$,P,1)="," THEN 1320
1300 IF P<LEN(M$) THEN 1280
1310 RETURN:REM ERROR
1320 UX=VAL(MID$(M$,4,P-1))
1330 AX=VAL(RIGHT$(M$,LEN(M$)-P))
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
```

```
1360 IF AC<1 OR AC>BREADTH THEN 1380
1370 Z$(UX,AX)=R$
1380 RETURN
1390 REM ************************************
1400 REM                  RANDOM
1410 AX=INT(RND(1)*BREADTH)
1420 UX=INT(RND(1)*DEPTH)
1430 Z$(UX,AX)=R$
1440 RETURN
1450 REM ************************************
1460 REM                  REPEAT
1470 M$=MID$(M$,4)
1480 IF ASC(M$)=69 THEN M$=MID$(M$,5)
1490 RECOUNT=VAL(M$)
1500 MARKER=PSN
1510 RETURN
1520 REM ************************************
1530 REM               END REPEAT
1540 RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ************************************
1580 REM                  FACE
1590 M$=MID$(M$,4)
1600 IF ASC(M$)=69 THEN M$=MID$(M$,3)
1610 NUM=VAL(M$)
1620 Y=INT((NUM+17.5)/45)*45
1630 IF Y=0 OR Y=360 THEN UP=-1:AC=0
1640 IF Y=45 THEN UP=-1:AC=1
1650 IF Y=90 THEN UP=0:AC=1
1660 IF Y=135 THEN UP=1:AC=1
1670 IF Y=180 THEN UP=1:AC=0
1680 IF Y=225 THEN UP=1:AC=-1
1690 IF Y=270 THEN UP=0:AC=-1
1700 IF Y=315 THEN UP=-1:AC=-1
1710 RETURN
1720 REM ************************************
1730 REM INITIALIZE
1740 CLS
1750 RANDOMIZE VAL(RIGHT$(TIME$,2))
1760 REM   ADJUST NEXT TWO LINES FOR
          BEST RESULTS ON YOUR SYSTEM
```

```
1770 BREADTH=80:REM CHARACTERS ACROSS
1780 DEPTH=24:REM CHARACTERS DOWN
1790 BREADTH=BREADTH-1
1800 DEPTH=DEPTH-3
1810 UP=-1:AC=0:REM STARTS FACING UP
1820 DIM A$[20]:REM FOR ROBOT PROGRAM
1830 DIM Z$[DEPTH,BREADTH]:REM DISPLAY
1840 T$="X":REM PUT SYMBOL HERE YOU
     WANT TO USE FOR ROBOT'S TRAIL
1850 AX=0:UX=0
1860 REM FILL ARRAY WITH SPACES
1870 FOR J=1 TO DEPTH
1880 FOR K=1 TO BREADTH
1890 Z$[J,K]=" "
1900 NEXT K
1910 NEXT J
1920 RETURN
```

Chapter Nine
The Point-Duty Robot

Now while the **Robot Logo** is interesting as far as it goes, I felt, once I had used it for a while, that it really didn't go far enough. The program was developed, like the others in this book, so they would run on just about any computer furnished with BASIC, and therefore it used an array to hold the robot's movement and current position. PRINTOUT triggered this array.

The *Point-Duty Robot* program makes use of the PSET command on my IBM PC to draw dots on a much finer resolution than the first program (32Ø by 2ØØ, rather than 8Ø by 24). Using a machine-specific command, of course, means that if a version of the program for your computer is not included in this book, you may have a little more trouble than usual in getting the program to run. However, the results are superb, and the effort involved in getting the program to run is sure to be repaid. If you don't have PSET on your computer to plot a point on the screen, check your computer's manual for SET, POINT, PLOT or a similar command. The Amstrad listings follow the IBM PC version. Spectrum BBC Micro and Commodore versions are in the appendices.

Here are the *Point-Duty Robot* listing, which is based on the *Robot Logo* program in the previous chapter. It has not been renumbered, so you should find it easy to modify the first listing into a second one. The commands are the same, except that PRINTOUT is no longer supported. It is not needed, as the program prints out automatically as it is running. (Note that line 1735 in the listing which follows is an IBM-specific one, to move the computer into graphics mode. You may need to drop or convert this line.)

```
100 REM POINT-DUTY ROBOT
110 GOSUB 1730:REM INITIALIZE
120 GOTO 490
```

```
130  REM  ********************************
140  REM
150  REM
160  REM
170  REM
180  REM
190  REM
200  REM
210  REM
220  REM
230  REM
240  REM
250  REM
260  REM
270  REM
280  REM
290  REM
300  REM
310  REM
320  DATA "*"
330  REM  ********************************
340  REM INT UX,AX
350  UX=INT(UX+.5):AX=INT(AX+.5)
360  RETURN
480  REM  ********************************
490  REM READ PROGRAM
500  COUNT=COUNT+1
510  READ A$(COUNT)
520  IF A$(COUNT)="*" THEN 550
530  IF COUNT<20 THEN 500
540  REM  ********************************
550  REM EXECUTE PROGRAM
560  PSN=0:REM PROGRAM STEP NUMBER
570  PSN=PSN+1
580  IF PSN=21 THEN 580:REM END
590  FLAG=0
600  M$=A$(PSN)
610  IF M$="*" THEN 610:REM END
620  N$=LEFT$(M$,2)
630  IF N$="ST" THEN 560:REM START AGAIN
650  IF N$="FO" THEN FLAG=1
```

```
660 IF N$="BA" THEN FLAG=2
670 IF N$="TU" THEN FLAG=3
680 IF N$="HO" THEN FLAG=4
700 IF N$="GO" THEN FLAG=5
710 IF N$="RA" THEN FLAG=6
720 IF N$="RE" THEN FLAG=7
730 IF N$="EN" THEN FLAG=8
740 IF N$="FA" THEN FLAG=9
750 ON FLAG GOSUB 780,940,1000,1160,1260
,1400,1460,1530,1580
760 GOTO 570
770 REM **************************
780 REM              FORWARD
790 M$=MID$(M$,4)
800 IF ASC(M$)=87 THEN M$=MID$(M$,6)
810 F$="F"
820 NUM=VAL(M$)
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN 880
850 IF AX<1 OR AX>BREADTH THEN 880
860  PRESET (AX,UX):PSET (AX,UX)
880 IF F$="F" THEN UX=UX+UP:AX=AX+AC
890 IF F$="B" THEN UX=UX-UP:AX=AX-AC
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM **************************
940 REM               BACK
950 M$=MID$(M$,4)
960 IF ASC(M$)=75 THEN M$=MID$(M$,3)
970 F$="B"
980 GOTO 820
990 REM **************************
1000 REM               TURN
1010 M$=MID$(M$,4)
1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
1030 NUM=VAL(M$)
1040 Y=INT((NUM+11.25)/22.5)
1050 IF Y=0 OR Y=16 THEN RETURN
1060 FOR J=1 TO Y
1065 IF UP=-2 AND AC=0 OR UP=2 AND AC=2
```

```
THEN AC=1:GOTO 1130
1070 IF UP=-2 AND AC=1 THEN AC=2:GOTO 11
30
1075 IF UP=-2 AND AC=2 OR UP=0 AND AC=-2
 THEN UP=-1:GOTO 1130
1080 IF UP=-1 AND AC=2 OR UP=1 AND AC=-2
 THEN UP=0:GOTO 1130
1085 IF UP=0 AND AC=2 OR UP=2 AND AC=-2
THEN UP=1:GOTO 1130
1090 IF UP=1 AND AC=2 THEN UP=2:GOTO 113
0
1095 IF UP=2 AND AC=1 THEN AC=0:GOTO 113
0
1100 IF UP=2 AND AC=0 THEN AC=-1:GOTO 11
30
1105 IF UP=2 AND AC=-1 THEN AC=-2:GOTO 1
130
1110 IF UP=-1 AND AC=-2 THEN UP=-2:GOTO
1130
1115 IF UP=-2 AND AC=-2 THEN AC=-1:GOTO
1130
1120 IF UP=-2 AND AC=-1 THEN AC=0
1130 NEXT J
1140 RETURN
1150 REM ********************************
1160 REM                HOME
1170 AX=INT[[BREADTH+.5]/2]
1180 UX=INT[[DEPTH+.5]/2]
1190 UP=-2:AC=0:REM FACES UP
1200 RETURN
1210 REM ********************************
1220 REM                CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ********************************
1260 REM              GO X,Y
1270 P=0
1280 P=P+1
1290 IF MID$[M$,P,1]="," THEN 1320
1300 IF P<LEN[M$] THEN 1280
1310 RETURN:REM ERROR
```

```
1320 UX=VAL[MID$[M$,4,P-1]]
1330 AX=VAL[RIGHT$[M$,LEN[M$]-P]]
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
1360 IF AC<1 OR AC>BREADTH THEN 1380
1370 PSET [AX,UX]
1380 RETURN
1390 REM ********************************
1400 REM                  RANDOM
1410 AX=INT[RND[1]*BREADTH]
1420 UX=INT[RND[1]*DEPTH]
1430 PSET[AX,UX]
1440 RETURN
1450 REM ********************************
1460 REM                  REPEAT
1470 M$=MID$[M$,4]
1480 IF ASC[M$]=69 THEN M$=MID$[M$,5]
1490 RECOUNT=VAL[M$]
1500 MARKER=PSN
1510 RETURN
1520 REM ********************************
1530 REM                END REPEAT
1540 RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ********************************
1580 REM                   FACE
1590 M$=MID$[M$,4]
1600 IF ASC[M$]=69 THEN M$=MID$[M$,3]
1610 NUM=VAL[M$]
1620 Y=INT[[NUM+11.25]/22.5]*22.5
1630 IF Y=0 OR Y=360 THEN UP=-2:AC=0
1635 IF Y=22.5 THEN UP=-2:AC=1
1640 IF Y=45 THEN UP=-2:AC=2
1645 IF Y=67.5 THEN UP=-1:AC=2
1650 IF Y=90 THEN UP=0:AC=2
1655 IF Y=112.5 THEN UP=1:AC=2
1660 IF Y=135 THEN UP=2:AC=2
1665 IF Y=157.5 THEN UP=2:AC=1
1670 IF Y=180 THEN UP=2:AC=0
1675 IF Y=202.5 THEN UP=2:AC=-1
```

```
1680 IF Y=225 THEN UP=2:AC=-2
1685 IF Y=247.5 THEN UP=1:AC=-2
1690 IF Y=270 THEN UP=0:AC=-2
1695 IF Y=292.5 THEN UP=-1:AC=-2
1700 IF Y=315 THEN UP=-2:AC=-2
1705 IF Y=337.5 THEN UP=-2:AC=-1
1710 RETURN
1720 REM **************************
1730 REM INITIALIZE
1735 SCREEN 1:REM THIS IS AN
     IBM-SPECIFIC GRAPHICS COMMAND
1740 CLS
1750 RANDOMIZE VAL(RIGHT$(TIME$,2))
1760 REM   ADJUST NEXT TWO LINES FOR
        BEST RESULTS ON YOUR SYSTEM
1770 BREADTH=320:REM CHARACTERS ACROSS
1780 DEPTH=200:REM CHARACTERS DOWN
1790 BREADTH=BREADTH-1
1800 DEPTH=DEPTH-3
1810 UP=-2:AC=0:REM STARTS FACING UP
1820 DIM A$(20):REM FOR ROBOT PROGRAM
1920 RETURN
```

Amstrad

```
100 REM POINT-DUTY ROBOT
105 REM AMSTRAD VERSION
110 GOSUB 1730:REM INITIALIZE
120 GOTO 490
130 REM ******************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ******************************
340 REM  INT UX,AX
350 UX=INT(UX+0.5):AX=INT(AX+0.5)
360 RETURN
370 REM ******************************
490 REM READ PROGRAM
500 COUNT=COUNT+1
510 READ A$(COUNT):A$(COUNT)=UPPER$(A$(C
OUNT))
520  IF  A$(COUNT)="*"  THEN 550
530  IF  COUNT<20 THEN 500
540 REM ******************************
550 REM EXECUTE PROGRAM
560 PSN=0:REM PROGRAM STEP NUMBER
570 PSN=PSN+1
```

```
580 IF PSN=21 THEN 580:REM END
590 FLAG=0
600 M$=A$(PSN)
610 IF M$="*" THEN 610:REM END
620 N$=LEFT$(M$,2)
630 IF N$="ST" THEN 560:REM START AGAIN
650 IF N$="FO" THEN FLAG=1
660 IF N$="BA" THEN FLAG=2
670 IF N$="TU" THEN FLAG=3
680 IF N$="HO" THEN FLAG=4
700 IF N$="GO" THEN FLAG=5
710 IF N$="RA" THEN FLAG=6
720 IF N$="RE" THEN FLAG=7
730 IF N$="EN" THEN FLAG=8
740 IF N$="FA" THEN FLAG=9
750 ON FLAG GOSUB 780,940,1000,1160,1260
,1400,1460,1530,1580
760 GOTO 570
770 REM ****************************
780 REM                FORWARD
790 M$=MID$(M$,4)
800 IF ASC(M$)=87 THEN M$=MID$(M$,6)
810 F$="F"
820 NUM=VAL(M$)
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN 880
850 IF AX<1 OR AX>BREADTH THEN 880
860 PLOT AX,UX,0:PLOT AX,UX,1
880 IF F$="F" THEN UX=UX+UP:AX=AX+AC
890 IF F$="B" THEN UX=UX-UP:AX=AX-AC
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ****************************
940 REM                BACK
950 M$=MID$(M$,4)
960 IF ASC(M$)=75 THEN M$=MID$(M$,3)
970 F$="B"
980 GOTO 820
990 REM ****************************
1000 REM               TURN
```

```
1010 M$=MID$(M$,4)
1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
1030 NUM=VAL(M$)
1040 Y=INT((NUM+11.25)/22.5)
1050 IF Y=0 OR Y=16 THEN RETURN
1060 FOR J=1 TO Y
1065 IF UP=-2 AND AC=0 OR UP=2 AND AC=2
THEN AC=1:GOTO 1130
1070 IF UP=-2 AND AC=1 THEN AC=2:GOTO 11
30
1075 IF UP=-2 AND AC=2 OR UP=0 AND AC=-2
 THEN UP=-1:GOTO 1130
1080 IF UP=-1 AND AC=2 OR UP=1 AND AC=-2
 THEN UP=0:GOTO 1130
1085 IF UP=0 AND AC=2 OR UP=2 AND AC=-2
THEN UP=1:GOTO 1130
1090 IF UP=1 AND AC=2 THEN UP=2:GOTO 113
0
1095 IF UP=2 AND AC=1 THEN AC=0:GOTO 113
0
1100 IF UP=2 AND AC=0 THEN AC=-1:GOTO 11
30
1105 IF UP=2 AND AC=-1 THEN AC=-2:GOTO 1
130
1110 IF UP=-1 AND AC=-2 THEN UP=-2:GOTO
1130
1115 IF UP=-2 AND AC=-2 THEN AC=-1:GOTO
1130
1120 IF UP=-2 AND AC=-1 THEN AC=0
1130 NEXT J
1140 RETURN
1150 REM ********************************
1160 REM              HOME
1170 AX=INT((BREADTH+0.5)/2)
1180 UX=INT(DEPTH+0.5)/2)
1190 UP=-2:AC=0:REM FACES UP
1200 RETURN
1210 REM ********************************
1220 REM              CLEAN
1230 GOSUB 1870
1240 RETURN
```

```
1250 REM ********************************
1260 REM              GO X,Y
1270 P=0
1280 P=P+1
1290 IF MID$(M$,P,1)="," THEN 1320
1300 IF P<LEN(M$) THEN 1280
1310 RETURN:REM ERROR
1320 UX=VAL(MID$(M$,4,P-1))
1330 AX=VAL(RIGHT$(M$,LEN(M$)-P))
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
1360 IF AC<1 OR AC>BREADTH THEN 1380
1370 PLOT AX,UX
1380 RETURN
1390 REM ********************************
1400 REM              RANDOM
1410 AX=INT(RND*BREADTH)
1420 UX=INT(RND*DEPTH)
1430 PLOT AX,UX
1440 RETURN
1450 REM ********************************
1460 REM              REPEAT
1470 M$=MID$(M$,4)
1480 IF ASC(M$)=69 THEN M$=MID$(M$,5)
1490 RECOUNT=VAL(M$)
1500 MARKER=PSN
1510 RETURN
1520 REM ********************************
1530 REM              END REPEAT
1540 RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ********************************
1580 REM              FACE
1590 M$=MID$(M$,4)
1600 F ASC(M$)=69 THEN M$=MID$(M$,3)
1610 NUM=VAL(M$)
1620 Y=INT((NUM+11.25)/22.5)*22.5
1630 IF Y=0 OR Y=360 THEN UP=-2:AC=0
1635 IF Y=22.5 THEN UP=-2:AC=1
1640 IF Y=45 THEN UP=-2:AC=2
```

```
1645 IF Y=67.5 THEN UP=-1:AC=2
1650 IF Y=90 THEN UP=0:AC=2
1655 IF Y=112.5 THEN UP=1:AC=2
1660 IF Y=135 THEN UP=2:AC=2
1665 IF Y=157.5 THEN UP=2:AC=1
1670 IF Y=180 THEN UP=2:AC=0
1675 IF Y=202.5 THEN UP=2:AC=-1
1680 IF Y=225 THEN UP=2:AC=-2
1685 IF Y=247.5 THEN UP=1:AC=-2
1690 IF Y=270 THEN UP=0:AC=-2
1695 IF Y=292.5 THEN UP=-1:AC=-2
1700 IF Y=315 THEN UP=-2:AC=-2
1705 IF Y=337.5 THEN UP=-2:AC=-1
1710 RETURN
1720 REM **************************
1730 REM INITIALIZE
1740 MODE 1:INK 0,0:INK 1,26:BORDER 0
1750 RANDOMIZE TIME
1770 BREADTH=319
1780 DEPTH=197
1810 UP=-2:AC=0:REM STARTS FACING UP
1820 DIM A$(20):REM FOR ROBOT PROGRAM
1920 RETURN
```

There are many enjoyable programs you can write for your little robot to follow. You can see from the FACE and TURN sections, the program supports angle changes of 17.5 degrees, rather than the coarser 45 degrees which was all the *Robot Logo* could cope with.

When you have *Point-Duty Robot* up and running, you might like to try the following programs.

The first one is GLASS MAGNOLIA:

```
GO 100,170
REPEAT 5
FORWARD 13
TURN 72
END REPEAT
START AGAIN
*
```

Adding a single, additional line to that program produces CHURCH WINDOW:

```
GO 100,170
REPEAT 4
FORWARD 13
TURN 90
END REPEAT
TURN 22.5
START AGAIN
*
```

HAMPTON COURT ROSE is an effective demonstration of the little robot in action:

```
GO 100,170
REPEAT 2
FORWARD 3
TURN 22.5
FORWARD 6
TURN 22.5
FORWARD 9
TURN 22.5
FORWARD 12
TURN 22.5
FORWARD 12
TURN 22.5
FORWARD 7
TURN 90
END REPEAT
START AGAIN
*
```

RCL continues to amaze with SPIRAL NEBULA:

```
GO 100,170
FORWARD 2
TURN 22.5
FORWARD 5
TURN 22.5
FORWARD 8
TURN 22.5
FORWARD 11
TURN 22.5
FORWARD 15
TURN 22.5
FORWARD 7
START AGAIN
*
```

CRAZY CRITTER decorates your screen with something surprising:

```
GO 100,170
REPEAT 2
FORWARD 10
TURN 90
FORWARD 12
END REPEAT
TURN 22.5
START AGAIN
*
```

And this is SIGNALMAN FREUD (who worked for the Vienna Railroad Company):

```
GO 100,170
TURN 135
FORWARD 25
REPEAT 4
FORWARD 10
TURN 90
END REPEAT
START AGAIN
*
```

In our next program, BUBBLES, we use two REPEAT loops. Note these are not, and cannot be, nested, as RCL does not support nested loops:

```
RANDOM
REPEAT 16
FORWARD 2
TURN 25
END REPEAT
RANDOM
REPEAT 16
FORWARD 1
TURN 25
END REPEAT
START AGAIN
*
```

Removing the second RANDOM line from BUBBLES turns the program into one I have called THE EYES ARE WATCHING YOU:

```
RANDOM
REPEAT 16
FORWARD 2
TURN 25
END REPEAT
REPEAT 16
FORWARD 1
TURN 25
END REPEAT
START AGAIN
*
```

From that we move to HAND ME DOWN MY DANCING CANE:

```
RANDOM
FORWARD 15
TURN 315
BACK 5
TURN 22.5
START AGAIN
*
```

And, finally, ONE-WAY STREET:

```
RANDOM
FORWARD 15
TURN 315
BACK 6
FORWARD 6
TURN 270
FORWARD 6
START AGAIN
*
```

I'd be very interested in seeing any programs you write in RCL, for either of the robot programs. You can get in touch with me via Interface Publications.

Chapter Ten
Simulating Intelligence

Artificial Intelligence (AI) is one of the 'hot' areas of computer science at the moment. An enormous amount of energy is being spent to get computers to behave in ways which appear intelligent. AI research has looked into such things as game playing. It has produced some highly creditable results as the current generation of chess machines bears witness. 'Expert systems', in which the computer uses the encoded expertise of human specialists to reach decisions and give advice, is a particularly fruitful area of AI development at present. Expert systems have been developed in a wide range of disciplines. Systems now in use can do such things as diagnose certain classes of diseases, assist in chemical synthesis, and help in the search for mineral deposits.

In this chapter, we'll concentrate on the game-playing side of artificial intelligence. The simulation we'll use plays a very good game of *Connect Four,* in which two players take it in turns to place pieces in the lowest available position in a column on a board, trying to get four of their pieces to form a line in any direction.

Game playing was one of the earliest areas of AI investigation which achieved significant success. Theorem proving programs, as well, performed well. The scope of AI research has broadened over the last 20 years and now covers such topics as perception (vision and speech), the understanding of human ('natural') languages, and the development of expert systems, mentioned briefly before.

It is interesting to note that computer programs which appear to display intelligence often work in ways quite different from the methods human beings bring to bear on similar problems. For example, chess masters do not play by systematically analysing tens of thousands of possible moves, ranking them hierarchically, and choosing the move which achieves the highest level. Instead,

a human player appears to make his or her decisions almost intuitively. Human experts, although they may make use of some or all of the raw data of an expert system working in the same field, do not generally reach their conclusions by consciously working through every piece of information which could possibly be relevant before deciding which of the potential outcomes has the highest probability of being correct.

Similarly, in our artificial intelligence simulation program, the computer works out its moves in a manner which is quite alien to our own thinking. Despite this, it plays extremely well, and will prove a difficult (almost impossible) opponent to defeat. It works by assigning a value to each possible move, and increments this value as it scans around the proposed move and in so doing finds another one of its pieces which could lead it towards victory. It then selects the move which has the highest total.

When you're making your moves, you're far more likely to just look at the board, note if there is any danger of the computer completing a row of four, and if so attempting to block it, or select a move which 'feels' as though it could help lead you to victory.

The lesson is clear. When writing your own artificial intelligence programs, look to the end result you want to achieve, rather than seek to emulate your own thinking processes in the situation.

Let's see our *Connect Four* program in action. You enter your move as a number between one and eight, and the computer automatically places your piece (a small letter 'o') at the *lowest* available position in the row designated by the number you have entered. The computer pieces are the letter 'M':

```
........        ........        ........
........        ........        ........
........        ........        ........
........        ........        ........
........        ........        ........
........        ........        ........
12345678        12345678        12345678

YOUR MOVE...    YOUR MOVE...    YOUR MOVE...
? 4             ? 6             ? 5
```

```
........
........
........
........
...Mo...
..MoMo..
12345678
```

```
YOUR MOVE...
? 6
```

```
........
........
........
.....M..
...Moo..
..MoMo..
12345678
```

```
YOUR MOVE...
? 7
```

```
........
........
........
.....M..
....MM..
...Mooo.
..MoMoo.
12345678
```

```
........
........
........
....MM..
...Moo..
..MoMoo.
12345678
```

```
YOUR MOVE...
? 7
```

```
I HAVE WON
```

The victory here is along the diagonal from the top of column six down to the bottom of column three.

Enter this program when you want to see an example of artificial intelligence in action:

```
10 REM CONNECT FOUR
20 REM A. W. PEARSON
30 CLS
40 PRINT
50 PRINT
60 PRINT "CONNECT FOUR"
70 PRINT
80 PRINT "ENTER YOUR MOVE AS A NUMBER BE
TWEEN"
90 PRINT "1 AND 8, ENTER 0 FOR A NEW GAM
E..."
100 FOR F=1 TO 1000:NEXT F
110 DIM A$[10,10],B[10,2]
120 FLAG=0
```

```
130 REM CHANGE NEXT LINE FOR YOUR OWN
      CHOICE OF SYMBOLS (C$-COMPUTER)
140 C$="M":H$="o":REM M FOR MACHINE!
150 FOR F=1 TO 8
160 B(F,1)=6
170 NEXT F
180 FOR F=1 TO 6
190 FOR G=1 TO 8
200 A$(F,G)="."
210 NEXT G
220 NEXT F
230 REM **************************
240 REM ACCEPT HUMAN MOVE
250 GOSUB 430
260 PRINT:PRINT "YOUR MOVE..."
270 INPUT A
280 IF A=0 THEN RUN
290 IF A<1 OR A>8 THEN 270
300 L=0
310 IF A$(L+1,A)<>"." OR L=6 THEN 340
320 L=L+1
330 GOTO 310
340 IF L=0 THEN 270
350 A$(L,A)=H$
360 B(A,1)=B(A,1)-1
370 GOSUB 430
380 GOSUB 560
390 GOSUB 430
400 GOTO 260
410 REM **************************
420 REM PRINT BOARD
430 CLS
440 FOR F=1 TO 6
450 FOR G=1 TO 8
460 PRINT A$(F,G);
470 NEXT G
480 PRINT
490 NEXT F
500 PRINT "12345678"
510 PRINT
520 IF FLAG=1 THEN PRINT "I HAVE WON":EN
D
```

```
530 RETURN
540 REM ****************************
550 REM COMPUTER MOVES
560 PRINT "MY MOVE..."
570 MV=0
580 FOR F=1 TO 8
590 B[F,2]=0
600 NEXT F
610 FOR F=1 TO 8
620 FOR X=-1 TO 1
630 FOR Y=-1 TO 1
640 IF B[F,1]=0 THEN 680
650 IF A$[B[F,1]+X,F+Y]="" OR A$[B[F,1]+
X,F+Y]="." THEN 680
660 IF A$[B[F,1]+X,F+Y]=H$ THEN GOSUB 81
0
670 IF A$[B[F,1]+X,F+Y]=C$ THEN GOSUB 91
0
680 NEXT Y
690 NEXT X
700 NEXT F
710 P=0
720 FOR F=1 TO 8
730 IF B[F,2]>P THEN P=B[F,2]:N=F
740 NEXT F
750 A$[B[N,1],N]=C$
760 B[N,1]=B[N,1]-1
770 N=0
780 P=0
790 RETURN
800 REM ****************************
810 MV=2
820 M1=MV
830 IF A$[B[F,1]+[X*2],F+[Y*2]]=H$ THEN
MV=MV+10
840 IF A$[B[F,1]-X,F-Y]=H$ THEN MV=MV+20
850 IF MV<>M1+10 THEN 870
860 IF A$[B[F,1]+[X*3],F+[Y*3]]=H$ THEN
MV=MV+1000
870 B[F,2]=B[F,2]+MV
880 M1=0
```

```
890 RETURN
900 REM ****************************
910 MV=2
920 M1=MV
930 IF A$[B[F,1]+[X*2],F+[Y*2]]=C$ THEN
MV=MV+9
940 IF A$[B[F,1]-X,F-Y]=C$ THEN MV=MV+20

950 IF MV<>M1+9 THEN 970
960 IF A$[B[F,1]+[X*3],F+[Y*3]]=C$ THEN
MV=MV+2000:FLAG=1
970 B[F,2]=B[F,2]+MV
980 RETURN
```

If you'd like to know more about artificial intelligence and expert systems, you might be interested in other books I have written on the subject:

Exploring Artificial Intelligence on your Microcomputer – (Interface Publications Ltd., UK, 1984) (Machine – specific versions available for Spectrum/Spectrum+, QL and BBC Micro)

Exploring Artificial Intelligence on your Commodore 64 – (Interface Publications Ltd., UK, 1985; Bantam Books, New York, 1985)

Exploring Expert Systems on your Microcomputer – (Interface Publications Ltd., UK, 1985)

Chapter Eleven
The Quevedo Chess Machine

In 1890, the first recorded attempt to produce a real machine which would play chess was made in Spain. The scientist Torres y Quevedo produced a little device which would play the end game of a king and rook against a king, taking the side with the rook. The machine was always able to force mate.

After reading about Quevedo's machine in David Levy's fascinating book *Chess and Computers* (Computer Science Press, Inc., Potomac, Maryland; 1976), I decided to try and write a program which would emulate the machine's behaviour. The machine could move in one of six possible ways, and this program tells you which of the six moves it has used in each case.

The moves include simply moving the rook away horizontally if the black king is in the same zone as the rook; and moving the rook one square horizontally, if the black king is not in the same zone as the rook, with the rook one square away, when the vertical distance between the two kings is two squares, and the number of squares they are apart horizontally is odd.

You'll find the computer makes a rapid assessment of the situation. Here it is in action. You are the single king (the $ sign), the computer's king is the "K" and its rook is the "R":

```
I USED MOVE 1

   ABCDEFGH
8 ........ 8
7 ........ 7
6 ........ 6
5 .R..K... 5
4 ........ 4
3 $....... 3
2 ........ 2
1 ........ 1

   ABCDEFGH

>> MOVE TO (LETTER, NO.)? A2

I USED MOVE 1

   ABCDEFGH
8 ........ 8
7 ........ 7
6 ........ 6
5 R...K... 5
4 ........ 4
3 ........ 3
2 $....... 2
1 ........ 1

   ABCDEFGH

>> MOVE TO (LETTER, NO.)? B2
```

```
I USED MOVE 1

   ABCDEFGH
8 ........ 8
7 ........ 7
6 ........ 6
5 .R..K... 5
4 ........ 4
3 ........ 3
2 .$...... 2
1 ........ 1

   ABCDEFGH

>> MOVE TO [LETTER, NO.]? C2

I USED MOVE 1

   ABCDEFGH
8 ........ 8
7 ........ 7
6 ........ 6
5 ..R.K... 5
4 ........ 4
3 ........ 3
2 ..$..... 2
1 ........ 1

   ABCDEFGH

>> MOVE TO [LETTER, NO.]? D2
```

```
I USED MOVE 1

     ABCDEFGH
   8 ........ 8
   7 ........ 7
   6 ........ 6
   5 ...RK... 5
   4 ........ 4
   3 ........ 3
   2 ...$.... 2
   1 ........ 1

     ABCDEFGH

>> MOVE TO (LETTER, NO.)? E2

I USED MOVE 2

     ABCDEFGH
   8 ........ 8
   7 ........ 7
   6 ........ 6
   5 ....K... 5
   4 ...R.... 4
   3 ........ 3
   2 ....$... 2
   1 ........ 1

     ABCDEFGH

>> MOVE TO (LETTER, NO.)? F2
```

```
I USED MOVE 1

   ABCDEFGH
8 ........ 8
7 ........ 7
6 ........ 6
5 ....K... 5
4 ....R... 4
3 ........ 3
2 .....$.. 2
1 ........ 1

   ABCDEFGH

>> MOVE TO (LETTER, NO.)? F1

I USED MOVE 1

   ABCDEFGH
8 ........ 8
7 ........ 7
6 ........ 6
5 ....K... 5
4 .....R.. 4
3 ........ 3
2 ........ 2
1 .....$.. 1

   ABCDEFGH

>> MOVE TO (LETTER, NO.)? Q

THANKS FOR THE GAME
```

Here's the listing so you can take on Senor Quevedo's machine for yourself:

```
10 REM QUEVEDO CHESS MACHINE
20 GOSUB 1510:REM INITIALISE
30 GOTO 60
40 GOSUB 1320:REM PRINT BOARD
50 GOSUB 110:REM COMPUTER MOVES
60 GOSUB 1320
70 GOSUB 1120:REM ACCEPT HUMAN MOVE
80 GOTO 40
90 END
100 REM ****************
110 REM COMPUTER MOVES
120 IF QUIT=1 THEN 1080
130 W1=WK
140 REM ****************
150 REM MOVE ONE
160 MOVE=1
170 KM=INT(BK/10)
180 RM=INT(R/10)
190 IF ABS(KM-RM)>3 THEN 330
200 A(R)=46
210 X=INT(BK/10):Y=INT(R/10)
220 IF X>Y THEN 270
230 IF A(R-10)<>46 THEN 270
240 IF A(R-19)=BK OR A(R-21)=BK OR A(R-2
0)=BK THEN 270
250 IF A(R-11)=BK OR A(R-9)=BK THEN 270
260 R=R-10:GOTO 300
270 IF A(R+10)<>46 THEN A(R)=R:GOTO 330
280 IF A(R+19)=BK OR A(R+21)=BK OR A(R+2
0)=BK THEN A(R)=R:GOTO 330
290 R=R+10
300 A(R)=ASC("R")
310 RETURN
320 REM ****************
330 REM MOVE TWO
340 MOVE=2
350 KM=BK-10*KM
360 RM=R-10*RM
```

```
370 IF ABS(KM-RM)<2 THEN 480
380 A(R)=46
390 IF R>11 THEN IF (A(R-12)=BK OR A(R-2
)=BK OR A(R+8)=BK) THEN A(R)=R:GOTO 480
400 IF R>11 THEN IF (A(R-1)=BK OR A(R-11
)=BK OR A(R+9)=BK) THEN A(R)=R:GOTO 480
410 Y=BK-10*INT(BK/10)
420 Z=R-10*INT(R/10)
430 IF (Z=1 OR Y>Z) AND A(R+1)=46 THEN R
=R+1:GOTO 450
440 R=R-1
450 A(R)=ASC("R")
460 RETURN
470 REM ****************
480 REM MOVE THREE
490 MOVE=3
500 WM=WK-10*INT(WK/10)
510 BM=BK-10*INT(BK/10)
520 IF ABS(WM-BM)<3 THEN 600
530 IF A(WK-1)<>46 OR A(WK-18)=BK OR A(W
K-2)=BK OR A(WK+8)=BK THEN 610
540 IF A(WK-11)=BK OR A(WK+9)=BK OR A(WK
-22)=BK THEN 610
550 A(WK)=46
560 WK=WK-1
570 A(WK)=ASC("K")
580 RETURN
590 REM ****************
600 REM MOVES FOUR, FIVE AND SIX
610 Z=ABS(INT(BK/10)-INT(WK/10))
620 IF Z=0 THEN 950
630 IF 2*INT(Z/2)=Z THEN 790
640 REM ****************
650 REM MOVE FOUR
660 MOVE=4
670 A(R)=46
680 IF A(R-10)<>46 THEN 720
690 IF A(R-9)=BK OR A(R-11)=BK THEN 720
700 IF A(R-19)=BK OR A(R-21)=BK OR A(R-2
0)=BK THEN 720
710 R=R-10:GOTO 760
```

```
720 IF A(R+10)<>46 THEN A(R)=R:GOTO 790
730 IF A(R+19)=BK OR A(R+21)=BK OR A(R+2
0)=BK THEN A(R)=ASC("R"):GOTO 790
740 IF A(R+11)=BK OR A(R+9)=BK THEN A(R)
=ASC("R"):GOTO 790
750 R=R+10
760 A(R)=ASC("R")
770 RETURN
780 REM ***************
790 REM MOVE FIVE
800 MOVE=5
810 J=INT(BK/10)
820 K=BK-10*J
830 L=INT(WK/10)
840 M=WK-10*L
850 Z=10:IF J<L THEN Z=-10
860 X=1:IF K<M THEN X=-1
870 A(WK)=46
880 W1=WK
890 WK=WK+Z+X
900 G=ABS(WK-BK)
910 IF G=1 OR G=9 OR G=10 OR G=11 THEN W
K=W1:A(WK)=75:GOTO 950
920 A(WK)=ASC("K")
930 RETURN
940 REM ***************
950 REM MOVE SIX
960 MOVE=6
970 A(R)=46
980 IF R>11 THEN IF A(R-12)=BK OR A(R-2)
=BK OR A(R+8)=BK OR A(R-1)<>46 THEN 1070

990 IF R>11 THEN IF (A(R-1)=BK OR A(R-11
)=BK OR A(R+9)=BK) THEN 1070
1000 Y=BK-10*INT(BK/10)
1010 Z=R-10*INT(R/10)
1020 IF (Z=1 OR Y>Z) AND A(R+1)=46 THEN
R=R+1:GOTO 1040
1030 R=R-1
1040 A(R)=ASC("R")
1050 RETURN
```

```
1060 REM **************
1070 GOSUB 1320
1080 PRINT:PRINT
1090 PRINT "I CONCEDE TO THE MASTER"
1100 END
1110 REM ******************
1120 REM ACCEPT HUMAN MOVE
1130 REM ENTER 'Q' TO QUIT
1140 MOVE=0
1150 PRINT ">> MOVE TO (LETTER, NO.)";
1160 INPUT G$
1170 IF G$="Q" THEN 1280
1180 IF LEN(G$)<>2 THEN 1160
1190 Z=ASC(G$)
1200 IF Z<65 OR Z>72 THEN 1160
1210 X=VAL(RIGHT$(G$,1))
1220 IF X<1 OR X>8 THEN 1160
1230 A(BK)=46
1240 BK=10*(Z-64)+X
1250 IF A(BK)=ASC("R") THEN QUIT=1
1260 A(BK)=ASC("$")
1270 RETURN
1280 PRINT:PRINT
1290 PRINT "THANKS FOR THE GAME"
1300 END
1310 REM ************
1320 REM PRINT BOARD
1330 CLS
1340 PRINT:PRINT
1350 IF MOVE>0 THEN PRINT "I USED MOVE"M
OVE
1360 IF MOVE=0 THEN PRINT
1370 PRINT:PRINT
1380 PRINT TAB(11);"ABCDEFGH"
1390 FOR J=8 TO 1 STEP -1
1400 PRINT TAB(8);J;
1410 FOR K=10 TO 80 STEP 10
1420 PRINT CHR$(A(J+K));
1430 NEXT K
1440 PRINT J
1450 NEXT J
```

```
1460 PRINT
1470 PRINT TAB(11);"ABCDEFGH"
1480 PRINT:PRINT
1490 RETURN
1500 REM ****************
1510 REM INITIALISATION
1520 CLS
1530 RANDOMIZE VAL(RIGHT$(TIME$,2))
1540 MOVE=0
1550 QUIT=0
1560 DIM A(130)
1570 FOR J=10 TO 80 STEP 10
1580 FOR K=1 TO 8
1590 A(J+K)=46:REM ASCII OF "."
1600 NEXT K
1610 NEXT J
1620 REM ** PLACE PIECES **
1630 REM BLACK KING - HUMAN
1640 BK=INT(RND(1)*3)+1
1650 BK=10*BK+BK+INT(RND(1)*5)
1660 A(BK)=ASC("$")
1670 REM WHITE KING - COMPUTER
1680 WK=INT(RND(1)*4)+4
1690 WK=10*WK+WK+INT(RND(1)*2)
1700 IF WK=BK THEN 1680
1710 A(WK)=ASC("K")
1720 REM WHITE ROOK - COMPUTER
1730 R=INT(RND(1)*4)+4
1740 R=10*R+R+INT(RND(1)*2)
1750 IF R=WK OR R=BK THEN 1730
1760 IF ABS(R-BK)<12 THEN 1730
1770 A(R)=ASC("R")
1780 RETURN
```

Chapter Twelve
Into the Political Arena

As we discussed earlier, computer programs can be used to simulate any process where the links between the elements of the process can be expressed as one or more mathematical relationships. This political simulation, *Washington D.C.,* shows this very clearly, with the mathematical relationships governing a grossly simplified version of the American economy.

You are the hapless occupant of the Oval Office, and – as you'll soon see – the economy is somewhat out of kilter. To retain the support of the people (and the presidency) you have to try and keep the whole thing together. You need to do your best to improve the standard of living, keep inflation and unemployment down, stimulate private investment and generally maintain an acceptable growth in the economy, all at same time. You'll soon discover a president's lot is not necessarily a happy one.

The simulation begins with a report to you on the state of the nation:

```
PRESIDENT HARTNELL:
YOUR ADMINISTRATION HAS BEEN IN
        POWER FOR .25 YEARS
-------------------------------
-----STATE OF THE NATION---------
-------------------------------
POPULATION 3000819
NO. UNEMPLOYED 300000      9 %
CURRENT WAGES $ 100   INFLATION 5 %
GOVT. EXPENDITURE LAST QTR. $M 118
UNEMPLOYMENT COST $M 0
INCOME FROM TAXES $M 0
BUDGET SURPLUS[+]/DEFICIT[-] $M 0
GROSS DOMESTIC PRODUCT $M 440
-------------------------------
PUBLIC INVESTMENT 0 Q 1 $M 236
-------------------------------
```

You start each run with a population of around 3,000,000 and an unemployment rate of about 10%. You are then asked to make some snap judgements on 'government spending', 'cost of wages' and on your administration's immigration policy:

```
OK, PRESIDENT HARTNELL...
ENTER GOVERNMENT SPENDING $M? 102
ENTER COST OF WAGES $M? 41
IS YOUR ADMINISTRATION IN FAVOR OF
IMMIGRATION THIS QUARTER [Y/N]?
                              OK...N
```

At the end of my second quarter in office, I found that while I had somehow kept inflation and unemployment under control, I was suddenly faced with a budget deficit of over $69 million! Public investment (and therefore, I hoped, public confidence) was on the increase however, up from $236 million at the end of the first quarter to more than $257 million at this point. I decide to increase spending slightly:

```
PRESIDENT HARTNELL:
YOUR ADMINISTRATION HAS BEEN IN
        POWER FOR .5 YEARS
---------------------------------
-----STATE OF THE NATION---------
---------------------------------
POPULATION 3001638
NO. UNEMPLOYED 287879     9 %
CURRENT WAGES $ 41   INFLATION 9 %
GOVT. EXPENDITURE LAST QTR. $M 102
UNEMPLOYMENT COST $M 11.8
INCOME FROM TAXES $M 44.4
BUDGET SURPLUS[+]/DEFICIT[-] $M-69.4
GROSS DOMESTIC PRODUCT $M 449.2
---------------------------------
PUBLIC INVESTMENT 0 Q 2 $M 275.7
---------------------------------
OK, PRESIDENT HARTNELL...
ENTER GOVERNMENT SPENDING $M? 107
ENTER COST OF WAGES $M? 43
```

I also decide to let a few of the world's tired and hungry masses into the land of milk and budget deficits:

```
IS YOUR ADMINISTRATION IN FAVOR OF
IMMIGRATION THIS QUARTER [Y/N]?
                          OK...Y
HOW MANY IMMIGRANTS WILL YOU ALLOW
          INTO THE US?
? 1000
```

At the end of my first year, the deficit has grown to £216 million, and standard of living has increased by 12%:

```
YOUR ADMINISTRATION HAS BEEN IN
         POWER FOR 1 YEARS
-------------------------------
-----STATE OF THE NATION---------
-------------------------------
POPULATION 3004277
NO. UNEMPLOYED 272309      9 %
CURRENT WAGES $ 46   INFLATION 9 %
GOVT. EXPENDITURE LAST QTR. $M 111
UNEMPLOYMENT COST $M 12.5
INCOME FROM TAXES $M 50.2
BUDGET SURPLUS[+]/DEFICIT[-] $M-216.7
GROSS DOMESTIC PRODUCT $M 486.2
CHANGE IN LIVING STANDARD 12 %
-------------------------------
PUBLIC INVESTMENT O Q 4 $M 364.4
-------------------------------
OK, PRESIDENT HARTNELL...
ENTER GOVERNMENT SPENDING $M? 120
ENTER COST OF WAGES $M? 65
IS YOUR ADMINISTRATION IN FAVOR OF
IMMIGRATION THIS QUARTER [Y/N]?
                          OK...N
```

I stumble on, narrowly avoiding crises each time I turn my head. Then, things start getting really hairy towards the end of my second year in office:

```
PRESIDENT HARTNELL:
YOUR ADMINISTRATION HAS BEEN IN
        POWER FOR 1.75 YEARS
-------------------------------
-----STATE OF THE NATION---------
-------------------------------
POPULATION 3008238
NO. UNEMPLOYED 377324     12 %
CURRENT WAGES $ 66  INFLATION 12 %
GOVT. EXPENDITURE LAST QTR. $M 125
UNEMPLOYMENT COST $M 24.9
INCOME FROM TAXES $M 69.4
BUDGET SURPLUS[+]/DEFICIT[-] $M-436.8
GROSS DOMESTIC PRODUCT $M 866
CHANGE IN LIVING STANDARD 13 %
-------------------------------
PUBLIC INVESTMENT 1 Q 3 $M 243.2
-------------------------------
OK, PRESIDENT HARTNELL...
ENTER GOVERNMENT SPENDING $M? 126
ENTER COST OF WAGES $M? 67
```

The outcome is inevitable. My time is up:

```
PRESIDENT HARTNELL, YOUR
ADMINISTRATION'S POOR ECONOMIC
PERFORMANCE HAS LED TO AN UNACCEPTABLE
RISE IN UNEMPLOYMENT
          AMONG OTHER THINGS...
-------------------------------
THE LACK OF CONFIDENCE IN YOUR
ADMINISTRATION IS SO BAD THERE ARE
CALLS FOR YOU TO RESIGN...YOU STEP
ASIDE TO ALLOW THE VICE-PRESIDENT TO
        OCCUPY THE OVAL OFFICE
-------------------------------
YOU WERE PRESIDENT FOR 2.25 YEARS
DURING YOUR TERM OF OFFICE, THE
POPULATION ROSE BY 9059
THE UNEMPLOYMENT RATE BECAME 17 %
AND THE INFLATION RATE BECAME 17.4 %
STANDARD OF LIVING CHANGED BY 17 %
```

```
AND THE BUDGET SURPLUS/DEFICIT
WAS $M-597.5
```

This is a program which you may well enjoy modifying. Change, for example, the income you get from taxes, to alter the simulation in a radical way. You can also fiddle with the thresholds at which the public start waving their fists.

Line 5Ø governs the increase in population from quarter to quarter. A steeper increase here will make it harder to keep the lid on unemployment.

I've been pretty generous with my use of REM statements in this program, so you should find it relatively easy to track down the variables you wish to modify. Here are the major ones, as listed in the initialization section of the program:

```
1150 REM ************************
1160 REM INITIALIZATION
1170 CLS
1180 RANDOMIZE VAL(RIGHT$(TIME$,2))
1190 ML=1000*1000
1200 P=3*ML:REM POPULATION
1210 U=P/10:REM UNEMPLOYMENT
1220 IV=236:REM INVESTMENT
1230 GE=118:REM GOVERNMENT EXPENDITURE
1240 GU=0:REM COST OF UNEMPLOYMENT
1250 GI=0:REM INCOME FROM TAXES
1260 WN=100:REM NEW WAGES
1270 WO=100:REM OLD WAGES
1280 IP=5:REM INFLATION PERCENT
1290 GDP=440:REM GROSS DOMESTIC PRODUCT
1300 AGDP=440:REM BASE YEAR GDP
1310 RGDP=440:REM REAL GDP
1320 CN=354:REM ECONOMIC CONSTANT
     (USED THROUGHOUT SIMULATION)
1330 Z=1:GAME=0:FLAG=0
1340 Y=0:REM YEAR
1350 PRINT "ENTER YOUR LAST NAME"
1360 INPUT A$
1370 RETURN
```

The part of the program which does all the calculating is fairly easy to unravel, once you know the principal variables:

```
490 REM ************************************
500 REM CALCULATIONS
510 CN=CN+(CN*IP/100)
520 U=P*(GE+IV)/(CN*10)+P*(IP/1000)
530 GU=U*WN/ML:REM UNEMPLOYMENT COST
540 GI=(((P-U)*WN*.4)/ML):REM INCOME
                          FROM TAXES
550 BD=BD+GI-GU-GE:REM BUDGET DEFICIT
560 AGDP=AGDP*(1+(IP/100))
570 GDP=GE+IV+((P-U)*WN/ML)
580 RGDP=GDP*440/AGDP
590 IP=((GE+IV)/CN*.1+(WN/WO)/100)*100
600 IV=(CN*67)/(IP*IP)
610 WO=WN
620 Z=Z+1:IF Z>4 THEN Z=1:Y=Y+1
630 RETURN
640 REM ************************************
650 REM CHECK BUDGET DEFICIT
660 IF BD>-1000 THEN RETURN
670 GAME=1
680 FLAG=1
690 RETURN
700 REM ************************************
710 REM CHECK STANDARD OF LIVING
720 IF Y<.75 THEN RETURN
730 IF INT((2*((RGDP/AGDP)*100)-100)/3)>
-15 THEN RETURN
740 GAME=1
750 FLAG=2
760 RETURN
770 REM ************************************
780 REM CHECK INFLATION RATE
790 IF IP<15 THEN RETURN
800 GAME=1
810 FLAG=3
820 RETURN
830 REM ************************************
840 REM CHECK UNEMPLOYMENT
```

```
850 IF INT(U*100/P)<15 THEN RETURN
860 GAME=1
870 FLAG=4
880 RETURN
890 REM ************************************
```

Here is the complete listing of the *Washington D.C.* program:

```
10 REM WASHINGTON D.C.
20 GOSUB 1160:REM INITIALIZE
30 REM ************************************
40 REM MAJOR CYCLE
50 P=INT(P+(P*273/ML))
60 GOSUB 160:REM PRINTOUT
70 GOSUB 510:REM CALCULATE
80 REM NOW CHECK END GAME
90 GOSUB 710:REM STANDARD OF LIVING
100 GOSUB 780:REM INFLATION RATE
110 GOSUB 840:REM UNEMPLOYMENT
120 IF GAME=1 THEN CLS:GOTO 890
130 GOTO 50
140 REM ************************************
150 REM PRINTOUT
160 CLS
170 PRINT "PRESIDENT ";A$;":"
180 PRINT "YOUR ADMINISTRATION HAS BEEN
IN                    POWER FOR"Y+Z/4"YEARS"

190 PRINT "----------------------------------
----"
200 PRINT "-----STATE OF THE NATION-----
----"
210 PRINT "----------------------------------
----"
220 PRINT "POPULATION"P
230 PRINT "NO. UNEMPLOYED"INT(U)"   "INT(
100*U/P)"%"
240 PRINT "CURRENT WAGES $"WO" INFLATION
"INT(IP)"%"
250 PRINT "GOVT. EXPENDITURE LAST QTR. $
M"GE
```

```
260 PRINT "UNEMPLOYMENT COST $M"INT(10*G
U)/10
270 PRINT "INCOME FROM TAXES $M"INT(GI*1
0)/10
280 PRINT "BUDGET SURPLUS(+)/DEFICIT(-)
$M"INT(BD*10)/10
290 PRINT "GROSS DOMESTIC PRODUCT $M"INT
(GDP*10)/10
300 IF Y+Z/4>.5 THEN PRINT "CHANGE IN LI
VING STANDARD"INT((2*((RGDP/AGDP)*100)-1
00)/3)"%"
310 PRINT "-----------------------------
----"
320 PRINT "PUBLIC INVESTMENT"Y"Q"Z"$M"IN
T(IV*10)/10
330 PRINT "-----------------------------
----"
340 PRINT "OK, PRESIDENT ";A$;"..."
350 INPUT "ENTER GOVERNMENT SPENDING $M"
;GE
360 INPUT "ENTER COST OF WAGES $M";WN
370 PRINT "IS YOUR ADMINISTRATION IN FAV
OR OF"
380 PRINT "IMMIGRATION THIS QUARTER (Y/N
)?"
390 X$=INKEY$
400 IF X$<>"Y" AND X$<>"N" THEN 390
410 PRINT TAB(20);"OK...";X$
420 FOR H=1 TO 1000:NEXT H
430 IF X$<>"Y" THEN RETURN
440 PRINT "HOW MANY IMMIGRANTS WILL YOU
ALLOW                INTO THE US?"
450 INPUT M
460 IF M<0 THEN 450
470 P=P+M
480 RETURN
490 REM **************************
500 REM CALCULATIONS
510 CN=CN+(CN*IP/100)
520 U=P*(GE+IV)/(CN*10)+P*(IP/1000)
530 GU=U*WN/ML:REM UNEMPLOYMENT COST
```

```
540  GI=[[[P-U]*WN*.4]/ML]:REM INCOME
                                FROM TAXES
550  BD=BD+GI-GU-GE:REM BUDGET DEFICIT
560  AGDP=AGDP*[1+[IP/100]]
570  GDP=GE+IV+[[P-U]*WN/ML]
580  RGDP=GDP*440/AGDP
590  IP=[[GE+IV]/CN*.1+[WN/WO]/100]*100
600  IV=[CN*67]/[IP*IP]
610  WO=WN
620  Z=Z+1:IF Z>4 THEN Z=1:Y=Y+1
630  RETURN
640  REM ****************************
650  REM CHECK BUDGET DEFICIT
660  IF BD>-1000 THEN RETURN
670  GAME=1
680  FLAG=1
690  RETURN
700  REM ****************************
710  REM CHECK STANDARD OF LIVING
720  IF Y<.75 THEN RETURN
730  IF INT[[2*[[RGDP/AGDP]*100]-100]/3]>
-15  THEN RETURN
740  GAME=1
750  FLAG=2
760  RETURN
770  REM ****************************
780  REM CHECK INFLATION RATE
790  IF IP<15 THEN RETURN
800  GAME=1
810  FLAG=3
820  RETURN
830  REM ****************************
840  REM CHECK UNEMPLOYMENT
850  IF INT[U*100/P]<15 THEN RETURN
860  GAME=1
870  FLAG=4
880  RETURN
890  REM ****************************
900  REM END OF THE GAME
910  PRINT "PRESIDENT ";A$;", YOUR"
920  PRINT "ADMINISTRATION'S POOR ECONOMI
C"
```

```
930 PRINT "PERFORMANCE HAS LED TO AN UNA
CCEPTABLE"
940 IF FLAG=1 THEN PRINT "BUDGET DEFICIT
"
950 IF FLAG=2 THEN PRINT "DROP IN THE ST
ANDARD OF LIVING"
960 IF FLAG=3 THEN PRINT "RISE IN THE IN
FLATION RATE"
970 IF FLAG=4 THEN PRINT "RISE IN UNEMPL
OYMENT"
980 PRINT "          AMONG OTHER THINGS..
."
990 PRINT "-----------------------------
----"
1000 PRINT "THE LACK OF CONFIDENCE IN YO
UR"
1010 PRINT "ADMINISTRATION IS SO BAD THE
RE ARE"
1020 PRINT "CALLS FOR YOU TO RESIGN...YO
U STEP"
1030 PRINT "ASIDE TO ALLOW THE VICE-PRES
IDENT TO"
1040 PRINT "          OCCUPY THE OVAL OFFIC
E"
1050 FOR H=1 TO 1000:NEXT H
1060 PRINT "-----------------------------
-----"
1070 PRINT "YOU WERE PRESIDENT FOR"Y+(Z*
.25)"YEARS"
1080 PRINT "DURING YOUR TERM OF OFFICE,
THE"
1090 PRINT "POPULATION ROSE BY"P-3*ML
1100 PRINT "THE UNEMPLOYMENT RATE BECAME
"INT(U*1000/P)/10"%"
1110 PRINT "AND THE INFLATION RATE BECAM
E"INT(10*IP)/10"%"
1120 PRINT "STANDARD OF LIVING CHANGED B
Y"INT((2*((RGDP/AGDP)*100)-100)/3)"%"
1130 PRINT "AND THE BUDGET SURPLUS/DEFIC
IT          WAS $M"INT(10*BD)/10
1140 END
```

```
1150 REM **************************
1160 REM INITIALIZATION
1170 CLS
1180 RANDOMIZE VAL(RIGHT$(TIME$,2))
1190 ML=1000*1000
1200 P=3*ML:REM POPULATION
1210 U=P/10:REM UNEMPLOYMENT
1220 IV=236:REM INVESTMENT
1230 GE=118:REM GOVERNMENT EXPENDITURE
1240 GU=0:REM COST OF UNEMPLOYMENT
1250 GI=0:REM INCOME FROM TAXES
1260 WN=100:REM NEW WAGES
1270 WO=100:REM OLD WAGES
1280 IP=5:REM INFLATION PERCENT
1290 GDP=440:REM GROSS DOMESTIC PRODUCT
1300 AGDP=440:REM BASE YEAR GDP
1310 RGDP=440:REM REAL GDP
1320 CN=354:REM ECONOMIC CONSTANT
     (USED THROUGHOUT SIMULATION)
1330 Z=1:GAME=0:FLAG=0
1340 Y=0:REM YEAR
1350 PRINT "ENTER YOUR LAST NAME"
1360 INPUT A$
1370 RETURN
```

Chapter Thirteen
Playing the Stock Market

The numbers a computer manipulates can represent anything, from cloud density, intricate subatomic reactions within a complex chemical experiment or the number of warriors taking part in a Roman raid on an English village. When computers were first used in business, the numbers generally stood for more mundane things such as statistics, sales charts and money. In the simulation in this chapter, we leave behind warriors in Roman Britain and revert to money.

You are a broker dealing in a small market of five companies. You start the simulation with $15,000 and are asked to enter a financial target for which you will be striving during the run:

```
ENTER YOUR GOAL FOR THIS SIMULATION,
       $16,000 TO $100,000
? 250000
TOO HIGH!

ENTER YOUR GOAL FOR THIS SIMULATION,
       $16,000 TO $100,000
? 16000
```

Once you've done that, the state of the market on day one is revealed:

```
------------------------------------------
DAY 1          YOUR GOAL IS  $ 16000
------------------------------------------

COMPANY NUMBER:
  1       2       3       4       5
CHANCE OF INCREASE (%):
  23      59      28      59      41
```

```
CHANCE OF DECREASE [%]:
  31      26      55        8       17
CURRENT VALUE PER SHARE:
$ 1.49 $ 1.99 $ 2.49  $ 2.99 $ 3.49
NO. OF SHARES HELD:
  2000   1500   1200    1000   800
BANK $ 265  TOTAL WORTH $ 15000
--------------------------------------
DO YOU WANT TO SELL ANY SHARES [Y/N]?
```

One cycle of buying and/or selling represents a day of trading. Each day, you are told the chance of each share rising or falling in price, and based on that information you can decide what to buy or sell:

```
--------------------------------------
DAY 1          YOUR GOAL IS  $ 16000
--------------------------------------
COMPANY NUMBER:
  1       2       3         4       5
CHANCE OF INCREASE [%]:
  23      59      28        59      41
CHANCE OF DECREASE [%]:
  31      26      55        8       17
CURRENT VALUE PER SHARE:
$ 1.49 $ 1.99 $ 2.49  $ 2.99 $ 3.49
NO. OF SHARES HELD:
  2000   1500   1200    1000   800
BANK $ 265  TOTAL WORTH $ 15000
--------------------------------------
WHICH ONES TO SELL?    OK 1
HOW MANY OF 1 TO SELL? 3000
YOU DON'T HAVE THAT MANY!
? 1500

--------------------------------------
DAY 1          YOUR GOAL IS  $ 16000
--------------------------------------
COMPANY NUMBER:
  1       2       3         4       5
CHANCE OF INCREASE [%]:
  23      59      28        59      41
```

```
CHANCE OF DECREASE (%):
  31      26      55        8       17
CURRENT VALUE PER SHARE:
$ 1.49 $ 1.99 $ 2.49  $ 2.99 $ 3.49
NO. OF SHARES HELD:
  500     1500    1200     1000    800
BANK $ 2500  TOTAL WORTH $ 15000
```

```
WHICH COMPANY TO BUY?    OK 4
HOW MANY OF 4 TO BUY? 750
```

You can only sell and buy shares once on each day, so you must choose wisely. After you've made your decisions, the computer will tell you how you rate to date as a stockbroker. Some of the assessments, as I'm sure you'll discover, can be quite harsh:

```
DAY 2          YOUR GOAL IS  $ 16000
```

```
COMPANY NUMBER:
  1       2       3         4       5
CHANCE OF INCREASE (%):
  51      11      43        32      35
CHANCE OF DECREASE (%):
  49      43      8         49      46
CURRENT VALUE PER SHARE:
$ 1.49 $ 2.1  $ 2.36  $ 3.16 $ 3.49
NO. OF SHARES HELD:
  600     1500    1200     1750    800
BANK $ 108  TOTAL WORTH $ 15329.06
```

```
YOUR RATING AFTER THAT ROUND OF
TRADING IS 'BAD'

    <PRESS SPACEBAR TO CONTINUE >
```

However, if you persevere, you may well prosper in the end. Just don't allow yourself to be discouraged by assessments such as 'hopeless':

```
DAY 8          YOUR GOAL IS  $ 16000
COMPANY NUMBER:
  1       2       3         4       5
CHANCE OF INCREASE (%):
  18      72      89        24      32
CHANCE OF DECREASE (%):
  4       25      1         55      50
CURRENT VALUE PER SHARE:
$ 1.53 $ 2.31 $ 2.57  $ 2.69 $ 3.68
NO. OF SHARES HELD:
  2100    400     1200     1750    800
BANK $ 191  TOTAL WORTH $ 15103.32
```

```
WHICH ONES TO SELL?    OK 1
HOW MANY OF 1 TO SELL? 2100
```

```
DAY 11         YOUR GOAL IS  $ 16000
COMPANY NUMBER:
  1       2       3         4       5
CHANCE OF INCREASE (%):
  87      30      18        27      30
CHANCE OF DECREASE (%):
  4       50      51        44      52
CURRENT VALUE PER SHARE:
$ 1.58 $ 2.44 $ 2.69  $ 2.65 $ 3.68
NO. OF SHARES HELD:
  600     0       2400     1750    800
BANK $ 355  TOTAL WORTH $ 15352.79
```

```
YOUR RATING AFTER THAT ROUND OF
TRADING IS 'HOPELESS'

    <PRESS SPACEBAR TO CONTINUE >
```

```
-------------------------------------------
DAY 17          YOUR GOAL IS  $ 16000
-------------------------------------------
COMPANY NUMBER:
  1       2       3         4       5
CHANCE OF INCREASE [%]:
  47      65      38        8       47
CHANCE OF DECREASE [%]:
  32      7       44        25      40
CURRENT VALUE PER SHARE:
$ 1.78 $ 2.5   $ 2.61   $ 2.98 $ 3.83
NO. OF SHARES HELD:
  0       0       2900     1750    800
BANK $ 229  TOTAL WORTH $ 16109.47
-------------------------------------------
YOU'VE HIT YOUR FINANCIAL GOAL!
```

The *Stock Market* listing is in the following chapter.

Chapter Fourteen
The Stock Market Listing

When you're ready to take on the brains of Wall Street, enter and run the following listing. Just don't be too ambitious with your early financial goals, or you'll never make it.

```
10 REM STOCK MARKET
20 CLS
30 RANDOMIZE VAL(RIGHT$(TIME$,2))
40 DIM S(5),N(5),P(5),D(5)
50 S(1)=1.49:S(2)=1.99:S(3)=2.49:S(4)=2.
99:S(5)=3.49
60 N(1)=2000:N(2)=1500:N(3)=1200:N(4)=10
00:N(5)=800
70 BB=265:TV=15000:QQ=15000:DAY=1
80 PRINT:PRINT "ENTER YOUR GOAL FOR THIS
 SIMULATION,"
90 PRINT TAB(8);"$16,000 TO $100,000"
100 INPUT GOAL
110 IF GOAL<16000 THEN PRINT "TOO LOW!":
GOTO 80
120 IF GOAL>100*1000 THEN PRINT "TOO HIG
H!":GOTO 80
130 REM ***************************
140 REM MAJOR LOOP
150 FOR C=1 TO 5
160 REM ADJUST THE 55 IN NEXT LINE TO
MODIFY GAME; 80 VERY HARD, 30 VERY EASY
170 D(C)=INT(RND(1)*55)+1
180 P(C)=INT(RND(1)*(100-D(C)))+1
190 NEXT C
200 GOSUB 230
210 GOTO 460
220 REM ***************************
```

```
230 REM PRINTOUT
240 CLS
250 PRINT "-------------------------------
------"
260 PRINT "DAY"DAY"          YOUR GOAL IS
$"GOAL
270 PRINT "-------------------------------
------"
280 PRINT "COMPANY NUMBER:"
290 PRINT TAB(2);1;TAB(9);2;TAB(16);3;TA
B(25);4;TAB(32);5
300 PRINT "CHANCE OF INCREASE (%):"
310 PRINT TAB(2);P(1);TAB(9);P(2);TAB(16
);P(3);TAB(25);P(4);TAB(32);P(5)
320 PRINT "CHANCE OF DECREASE (%):"
330 PRINT TAB(2);D(1);TAB(9);D(2);TAB(16
);D(3);TAB(25);D(4);TAB(32);D(5)
340 PRINT "CURRENT VALUE PER SHARE:"
350 PRINT "$";INT(S(1)*100)/100;TAB(8);"
$";INT(S(2)*100)/100;
360 PRINT TAB(15);"$";INT(S(3)*100)/100;
TAB(23);"$";INT(S(4)*100)/100;
370 PRINT TAB(30);"$";INT(S(5)*100)/100
380 PRINT "NO. OF SHARES HELD:"
390 PRINT TAB(2);N(1);TAB(9);N(2);TAB(16
);N(3);TAB(24);N(4);TAB(31);N(5)
400 PRINT "BANK $"INT(BB)" TOTAL WORTH $
"TV
410 PRINT "-------------------------------
------"
420 IF TV>GOAL THEN PRINT "YOU'VE HIT YO
UR FINANCIAL GOALI":END
430 RETURN
440 REM **************************
450 REM          ** SELL **
460 PRINT "DO YOU WANT TO SELL ANY SHARE
S [Y/N]?"
470 A$=INKEY$
480 IF A$<>"Y" AND A$<>"N" THEN 470
490 IF A$="N" THEN 690
500 GOSUB 230
```

```
510 PRINT "WHICH ONES TO SELL?";
520 A$=INKEY$
530 IF A$<"1" OR A$>"5" THEN 520
540 C=VAL[A$]
550 PRINT "     OK"C
560 PRINT "HOW MANY OF"C"TO SELL";
570 INPUT N
580 IF N>N[C] THEN PRINT "YOU DON'T HAVE
 THAT MANY!":GOTO 570
590 REM ************************
600 REM ADJUST FIGURES AFTER SALE
610 BB=BB+S[C]*N:REM ADD VALUE TO BANK
620 N[C]=N[C]-N:REM SUBTRACT NO. SOLD
630 TV=0:REM SET TOTAL WORTH TO ZERO
640 REM NOW DETERMINE CURRENT WORTH
650 FOR C=1 TO 5
660 TV=TV+N[C]*S[C]
670 NEXT C
680 TV=TV+BB:REM ADD IN BANK BALANCE
690 GOSUB 230
700 REM ************************
710 REM            ** BUY **
720 PRINT "DO YOU WANT TO BUY ANY SHARES
 [Y/N]?"
730 A$=INKEY$
740 IF A$<>"Y" AND A$<>"N" THEN 730
750 IF A$="N" THEN 890
760 GOSUB 230
770 PRINT "WHICH COMPANY TO BUY?";
780 A$=INKEY$
790 IF A$<"1" OR A$>"5" THEN 780
800 C=VAL[A$]
810 PRINT "     OK"C
820 PRINT "HOW MANY OF"C"TO BUY";
830 INPUT N
840 IF N*S[C]>BB THEN PRINT "YOU DON'T H
AVE ENOUGH MONEY!":GOTO 830
850 REM ************************
860 REM ADJUST FIGURES AFTER BUY
870 BB=BB-S[C]*N
880 N[C]=N[C]+N
```

```
890  TV=0
900  FOR C=1 TO 5
910  TV=TV+N[C]*S[C]
920  NEXT C
930  TV=TV+BB
940  GOSUB 230
950  REM ************************
960  REM  MODIFY ALL INDICATORS
970  TV=0
980  FOR C=1 TO 5
990  K=INT[RND[1]*100]+1
1000  IF K<P[C] THEN S[C]=S[C]*[1+[P[C]/1
000]]
1010  K=INT[RND[1]*100]+1
1020  IF K<D[C] THEN S[C]=S[C]/[1+[D[C]/1
000]]
1030  TV=TV+[S[C]*N[C]]
1040  NEXT C
1050  TV=TV+BB
1060  QQ=QQ*1.005
1070  W=[TV*100/QQ]-100
1080  IF W=0 THEN W=.1
1090  W=W+6
1100  IF W<1 THEN W=1
1110  IF W>15 THEN W=15
1120  RESTORE
1130  FOR T=1 TO W
1140  READ A$
1150  NEXT T
1160  PRINT
1170  REM ************************
1180  REM GIVE RATING, START NEW ROUND
1190  PRINT "YOUR RATING AFTER THAT ROUND
 OF"
1200  PRINT "TRADING IS '";A$;"'"
1210  PRINT:PRINT "   <PRESS SPACEBAR TO
CONTINUE >"
1220  IF INKEY$<>" " THEN 1220:REM NOTE
            SPACE BETWEEN QUOTE MARKS
1230  DAY=DAY+1
1240  GOTO 150
```

```
1250 DATA "HOPELESS","VERY, VERY POOR"
1260 DATA "TERRIBLE","AWFUL","BAD"
1270 DATA "VERY ORDINARY","AVERAGE"
1280 DATA "REASONABLE","A LITTLE ABOVE A
VERAGE"
1290 DATA "FAIRLY GOOD","GOOD","VERY GOO
D"
1300 DATA "GREAT","EXCELLENT","SUPERLATI
VE"
```

If you want to modify the degree of difficulty of this simulation, change the 55 in line 17Ø. The lower the number, down to 3Ø, the easier the program is to run with a profit. The higher the number, up to 8Ø, the more difficult it becomes. The simulation will not run even remotely realistically if you substitute values less than 30, or greater than 8Ø, in line 17Ø.

Chapter Fifteen
Running A Vehicle Factory

Some of the complex processes of manufacturing and sales are presented in this simulation (albeit in a very simplified form). It took 6Ø years for the car market to become saturated in the States. You have been unfortunate enough to be appointed to the number one spot at a major car manufacturing concern the very month the saturation finally became general knowledge.

To save the day (and your reputation as a hard-hitting, straight-from-the-shoulder, trouble-shootin', crisis-solvin' executive) you must put the tottering firm back on its feet.

You take over just as the accountants announce the company is losing $6Ø million a year. Your task, needless to say, is to try and make the company profitable.

Here's how it begins:

```
INDUSTRY SALES 50000 IN MONTH 0
-------------------------------
YOU HAVE 12000 EMPLOYEES
AVERAGE WAGES ARE $ 22995
  OR $M 22.9 PER MONTH
-------------------------------
```

As you can see, the total sales within the vehicle industry for the month before the simulation began (month Ø) were 5Ø,ØØØ. You start the simulation with your company holding about a quarter of the total industry sales, around 12,5ØØ vehicles.

You have four factories, currently producing more vehicles than you can sell:

```
----------------------------------
MAXIMUM MONTHLY OUTPUT:
   FACTORY 1: 8900
   FACTORY 2: 3250
   FACTORY 3: 2500
   FACTORY 4: 1625
----------------------------------
TOTAL OUTPUT IS 16275
----------------------------------
EFFICIENCY LEVEL IS 77 %
DO YOU WANT TO EXPAND OUTPUT [Y/N]?
                                N
DO YOU WANT TO SELL FACTORY 4 [Y/N]?
                                N
```

The average wage for each employee is around $23,000 when the simulation begins. This will rise each three months, due to unions demanding parity be kept with inflation. Although you cannot reduce wages, or refuse to include the rise due to inflation, you can decide how many people to hire or fire, and at what price you will sell each vehicle (when the simulation begins, each car is selling for around $12,000):

```
INDUSTRY SALES 50000 IN MONTH 0
----------------------------------
YOU HAVE 12000 EMPLOYEES
AVERAGE WAGES ARE $ 22995
 OR $M 22.9 PER MONTH
----------------------------------
HOW MANY EMPLOYEES TO HIRE? 0
HOW MANY EMPLOYEES TO FIRE? 567
----------------------------------
YOU HAVE 11433 EMPLOYEES
AVERAGE WAGES ARE $ 22995
 OR $M 21.9 PER MONTH
----------------------------------
WHAT IS YOUR SELLING PRICE? 12345
```

Once you've answered all the relevant questions, the simulation reports back to you on how well (or otherwise) you have performed:

```
INDUSTRY SALES 48833 IN MONTH 1
YOUR SALES: 13386   [ 27.4 % OF TOTAL ]
---------------------------------
YOU HAVE 11433 EMPLOYEES
AVERAGE WAGES ARE $ 22995
  OR $M 21.9 PER MONTH
---------------------------------
AVERAGE COST PER VEHICLE  IS $ 10192
AND AVERAGE SELLING PRICE IS $ 12345
SO THE AVERAGE PROFIT IS $ 2152
OR $M 28.8 PER MONTH
---------------------------------
PROFIT FOR THE MONTH IS $M 6.9
& TOTAL PROFIT TO DATE IS $M 6.9
---------------------------------
DO YOU WANT TO RESIGN [Y/N]?
                              N
```

That wasn't too bad. You actually made a profit. The process continues:

```
INDUSTRY SALES 48833 IN MONTH 1
YOUR SALES: 13386   [ 27.4 % OF TOTAL ]
---------------------------------
YOU HAVE 10233 EMPLOYEES
AVERAGE WAGES ARE $ 22995
  OR $M 19.6 PER MONTH
---------------------------------
AVERAGE COST PER VEHICLE  IS $ 10192
AND AVERAGE SELLING PRICE IS $ 12345
SO THE AVERAGE PROFIT IS $ 2152
OR $M 28.8 PER MONTH
---------------------------------
PROFIT FOR THE MONTH IS $M 6.9
& TOTAL PROFIT TO DATE IS $M 6.9
---------------------------------
WHAT IS YOUR SELLING PRICE? 24000
TOO BIG A CHANGE FOR THE MARKET
WHAT IS YOUR SELLING PRICE? 14000
```

```
INDUSTRY SALES 51485 IN MONTH 2
YOUR SALES: 6435   [ 12.5 % OF TOTAL ]
-----------------------------------
YOU HAVE 10233 EMPLOYEES
AVERAGE WAGES ARE $ 22995
  OR $M 19.6 PER MONTH
-----------------------------------
AVERAGE COST PER VEHICLE   IS $ 11630
AND AVERAGE SELLING PRICE IS $ 14000
SO THE AVERAGE PROFIT IS $ 2369
OR $M 15.2 PER MONTH
-----------------------------------
PROFIT FOR THE MONTH IS $M-4.4
& TOTAL PROFIT TO DATE IS $M 2.5
-----------------------------------
DO YOU WANT TO RESIGN [Y/N]?
                              N
```

And just when you think you're getting it all under control, inflation exacts its toll:

```
INFLATION RATE THIS QUARTER IS 2.5 %
AVERAGE WAGES BILL WILL NOW RISE TO
        $ 23569    PER EMPLOYEE

                ANY KEY TO CONTINUE
```

Your next month is not very good:

```
INDUSTRY SALES 49184 IN MONTH 3
YOUR SALES: 6837   [ 13.9 % OF TOTAL ]
-----------------------------------
YOU HAVE 9743 EMPLOYEES
AVERAGE WAGES ARE $ 23569.88
  OR $M 19.1 PER MONTH
-----------------------------------
AVERAGE COST PER VEHICLE   IS $ 11835
AND AVERAGE SELLING PRICE IS $ 14500
SO THE AVERAGE PROFIT IS $ 2664
OR $M 18.2 PER MONTH
```

```
--------------------------------
PROFIT FOR THE MONTH IS $M-1
& TOTAL PROFIT TO DATE IS $M 1.5
--------------------------------
DO YOU WANT TO RESIGN [Y/N]?
```

You start the simulation, as you've seen, with four factories. You have the option, at any time, of selling off factory number four. Alternatively, you can increase the output of the factories. You'll find that the minimum cost per vehicle (and therefore the maximum profit per sale) occurs when your factories are running at 85% efficiency. We decide to sell off factory four, to let the income from that sale offset our losses:

```
YOUR MONTHLY SALES ARE 5517
--------------------------------
.MAXIMUM MONTHLY OUTPUT:
   FACTORY 1: 8900
   FACTORY 2: 3250
   FACTORY 3: 2500
   FACTORY 4: 1625
--------------------------------
TOTAL OUTPUT IS 16275
--------------------------------
EFFICIENCY LEVEL IS 33 %
DO YOU WANT TO EXPAND OUTPUT [Y/N]?
                              N
DO YOU WANT TO SELL FACTORY 4 [Y/N]?
                              Y
FACTORY 4 IS VALUED FOR SALE AT $M104
YOU CAN'T REBUY IT LATER IF
YOU SELL IT...
DO YOU WANT TO SELL [Y/N]?
                              Y
```

I've tried this simulation several times. Each time, the same fate occurs. Perhaps I'm just not cut out to run a multi-million dollar business.

```
INDUSTRY SALES 48991 IN MONTH 10
YOUR SALES: 5727   [ 11.6 % OF TOTAL ]
--------------------------------
YOU HAVE 9703 EMPLOYEES
```

```
AVERAGE WAGES ARE $ 24521.51
 OR $M 19.8 PER MONTH
---------------------------------
AVERAGE COST PER VEHICLE  IS $ 12418
AND AVERAGE SELLING PRICE IS $ 15400
SO THE AVERAGE PROFIT IS $ 2981
OR $M 17 PER MONTH
---------------------------------
PROFIT FOR THE MONTH IS $M-2.8
& TOTAL PROFIT TO DATE IS $M 68.4
---------------------------------
DO YOU WANT TO RESIGN [Y/N]?


YOU JUST MADE YOUR TWELFTH MONTHLY
LOSS IN A ROW.....................
      YOUR EMPLOYMENT
      IS HEREBY TERMINATED!!
```

A much better outcome is, however, possible for those who are more skilled than I am:

```
WELL DONE! THE COMPANY HAS MADE MORE
   THAN $M200.    YOU'VE BEEN MADE
         A MEMBER OF THE BOARD
```

Here are the principal variables, as they are assigned at the start of a run of this simulation:

```
1630 REM ************************
1640 REM INITIALIZATION
1650 CLS
1660 RANDOMIZE VAL[RIGHT$[TIME$,2]]
1670 DIM M[5],Y[5]
1680 NE=12000:REM STARTING NO EMPLOYEES
1690 AW=22995:REM STARTING AVERAGE WAGE
1700 AC=11100:REM COST PRICE/VEHICLE
1710 AS=12000:REM SELLING PRICE
1720 MI=50*1000:MC=10100
1730 Y[3]=12500
1740 MS=25:EF=77:FA=160:SF=0:MT=0
```

```
1750 FOR J=1 TO 5
1760 READ M[J]
1770 NEXT J
1780 RETURN
1790 DATA 8900,3250,2500,1625,16275
```

As you can see, the REM statements allow you to trace the various formulae which are used in this simulation:

```
220 INPUT "HOW MANY EMPLOYEES TO HIRE";H
E
230 NE=NE+HE:IF HE>0 THEN 260
240 INPUT "HOW MANY EMPLOYEES TO FIRE";H
E
250 NE=NE-HE
260 GOSUB 650
270 P1=AS:REM SET P1 EQUAL TO OLD PRICE
280 INPUT "WHAT IS YOUR SELLING PRICE";A
S
290 REM NEXT LINE REJECTS TOO BIG A
          CHANGE IN SELLING PRICE
300 IF ABS(P1-AS)>2500 THEN PRINT "TOO B
IG A CHANGE FOR THE MARKET":GOTO 280
310 CLS
320 PRINT:PRINT:PRINT
330 MI=INT(RND(1)*4000)+48*1000:REM THIS
      MONTH'S SALES BY INDUSTRY
340 C=C+1:REM COUNTS NUMBER OF MONTHS
350 IF C<3 THEN 470
360 M=INT(RND(1)*10+1)/4:REM INFLATION
370 CLS
380 PRINT "INFLATION RATE THIS QUARTER I
S"M"%"
390 PRINT "AVERAGE WAGES BILL WILL NOW R
ISE TO"
400 AW=(AW*M/100)+AW
410 PRINT TAB(8);"$"INT(AW)"  PER EMPLOY
EE"
420 IF INKEY$<>"" THEN 420
430 PRINT:PRINT TAB(12);"ANY KEY TO CONT
INUE"
```

```
440 IF INKEY$="" THEN 440
450 FA=(FA*M/100)+FA
460 C=0
470 Y(1)=NE*15/12:REM SALES BASED ON
                NUMBER OF EMPLOYEES
480 Y(2)=(100-AS/FA)*MI/100:REM SALES
  BASED ON MONTHLY INDUSTRY SALES
490 REM NEXT LINES SET LOWEST FIGURE
    FROM Y(1),Y(2),M(5) EQUAL TO Y(3)
500 IF Y(1)<Y(2) AND Y(1)<M(5) THEN Y(3)
=Y(1):GOTO 540
510 IF Y(2)<Y(1) AND Y(2)<M(5) THEN Y(3)
=Y(2):GOTO 540
520 Y(3)=M(5)
530 REM NEXT LINES DETERMINE
          MONTHLY SALES
540 IF ABS(P1-AS)<501 THEN Y(3)=3.6*Y(3)
/3
550 IF Y(3)>M(5) THEN Y(3)=Y(3)-1975:GOT
O 550
560 MC=(MC*M/100)+MC
570 EF=Y(3)/M(5)*100:REM EFFICIENCY %
 AS SALES DIVIDED BY TOTAL OUTPUT
580 AC=(MC*(ABS(85-EF)/3)/100)+MC:REM
       AVERAGE COST PER VEHICLE
590 MP=((Y(3)*(AS-AC))-(NE*AW/12)):REM
       MONTHLY PROFIT
600 MP=INT(MP/(100*1000))
610 TP=TP+MP/10:REM TOTAL PROFIT
                    IN MILLIONS
620 M=0
```

The complete listing for *Detroit City* is in the next chapter.

Chapter Sixteen
The Detroit City Listing

Here's the complete listing of the *Detroit City* program:

```
10 REM DETROIT CITY
20 GOSUB 1640:REM INITIALIZE
30 GOTO 110
40 MT=MT+1:REM COUNTS MONTHS
50 GOSUB 650
60 IF TP>200 THEN 1560
70 PRINT "DO YOU WANT TO RESIGN (Y/N)?"
80 GOSUB 1010
90 IF A$="Y" THEN PRINT "OK, CHIEF":END
100 GOSUB 1380
110 GOSUB 650
120 FOR T=1 TO 1000:NEXT T
130 GOSUB 850
140 PRINT "DO YOU WANT TO EXPAND OUTPUT
(Y/N)?"
150 GOSUB 1010
160 IF A$="Y" THEN 1080
170 IF SF=1 THEN 210
180 PRINT "DO YOU WANT TO SELL FACTORY 4
 (Y/N)?"
190 GOSUB 1010
200 IF A$="Y" THEN 1250
210 GOSUB 650
220 INPUT "HOW MANY EMPLOYEES TO HIRE";H
E
230 NE=NE+HE:IF HE>0 THEN 260
240 INPUT "HOW MANY EMPLOYEES TO FIRE";H
E
250 NE=NE-HE
260 GOSUB 650
```

```
270 P1=AS:REM SET P1 EQUAL TO OLD PRICE
280 INPUT "WHAT IS YOUR SELLING PRICE";A
S
290 REM NEXT LINE REJECTS TOO BIG A
          CHANGE IN SELLING PRICE
300 IF ABS(P1-AS)>2500 THEN PRINT "TOO B
IG A CHANGE FOR THE MARKET":GOTO 280
310 CLS
320 PRINT:PRINT:PRINT
330 MI=INT(RND(1)*4000)+48*1000:REM THIS
      MONTH'S SALES BY INDUSTRY
340 C=C+1:REM COUNTS NUMBER OF MONTHS
350 IF C<3 THEN 470
360 M=INT(RND(1)*10+1)/4:REM INFLATION
370 CLS
380 PRINT "INFLATION RATE THIS QUARTER I
S"M"%"
390 PRINT "AVERAGE WAGES BILL WILL NOW R
ISE TO"
400 AW=(AW*M/100)+AW
410 PRINT TAB(8);"$"INT(AW)"  PER EMPLOY
EE"
420 IF INKEY$<>"" THEN 420
430 PRINT:PRINT TAB(12);"ANY KEY TO CONT
INUE"
440 IF INKEY$="" THEN 440
450 FA=(FA*M/100)+FA
460 C=0
470 Y(1)=NE*15/12:REM SALES BASED ON
                NUMBER OF EMPLOYEES
480 Y(2)=(100-AS/FA)*MI/100:REM SALES
  BASED ON MONTHLY INDUSTRY SALES
490 REM NEXT LINES SET LOWEST FIGURE
    FROM Y(1),Y(2),M(5) EQUAL TO Y(3)
500 IF Y(1)<Y(2) AND Y(1)<M(5) THEN Y(3)
=Y(1):GOTO 540
510 IF Y(2)<Y(1) AND Y(2)<M(5) THEN Y(3)
=Y(2):GOTO 540
520 Y(3)=M(5)
530 REM NEXT LINES DETERMINE
          MONTHLY SALES
```

```
540 IF ABS(P1-AS)<501 THEN Y(3)=3.6*Y(3)
/3
550 IF Y(3)>M(5) THEN Y(3)=Y(3)-1975:GOT
O 550
560 MC=(MC*M/100)+MC
570 EF=Y(3)/M(5)*100:REM EFFICIENCY %
 AS SALES DIVIDED BY TOTAL OUTPUT
580 AC=(MC*(ABS(85-EF)/3)/100)+MC:REM
      AVERAGE COST PER VEHICLE
590 MP=((Y(3)*(AS-AC))-(NE*AW/12)):REM
      MONTHLY PROFIT
600 MP=INT(MP/(100*1000))
610 TP=TP+MP/10:REM TOTAL PROFIT
                    IN MILLIONS
620 M=0
630 GOTO 40
640 REM **************************
650 REM REPORT PRINTOUT
660 CLS
670 PRINT "INDUSTRY SALES"MI"IN MONTH"MT
680 IF MT>0 THEN PRINT "YOUR SALES:"INT(
Y(3))" ("INT(Y(3)*1000/MI)/10"% OF TOTAL
 )"
690 PRINT "-------------------------
---"
700 PRINT "YOU HAVE"NE"EMPLOYEES"
710 PRINT "AVERAGE WAGES ARE $"AW
720 PRINT " OR $M"INT(AW*NE/(100*1000)/1
2)/10"PER MONTH"
730 PRINT "-------------------------
---"
740 IF MT=0 THEN RETURN
750 PRINT "AVERAGE COST PER VEHICLE  IS
$"INT(AC)
760 PRINT "AND AVERAGE SELLING PRICE IS
$"INT(AS)
770 PRINT "SO THE AVERAGE PROFIT IS $"IN
T(AS-AC)
780 PRINT "OR $M"INT((AS-AC)*Y(3)/(100*1
000))/10"PER MONTH"
```

```
790 PRINT "-----------------------------
---"
800 PRINT "PROFIT FOR THE MONTH IS $M"MP
/10
810 PRINT "& TOTAL PROFIT TO DATE IS $M"
INT(TP*10)/10
820 PRINT "-----------------------------
---"
830 RETURN
840 REM **************************
850 REM MONTH REPORT
860 CLS
870 IF MT>0 THEN PRINT "YOUR MONTHLY SAL
ES ARE"INT(Y(3))
880 PRINT "-----------------------------
---"
890 PRINT "MAXIMUM MONTHLY OUTPUT:"
900 PRINT TAB(3);"FACTORY 1:"INT(M(1))
910 PRINT TAB(3);"FACTORY 2:"INT(M(2))
920 PRINT TAB(3);"FACTORY 3:"INT(M(3))
930 IF SF=1 THEN 960
940 PRINT TAB(3);"FACTORY 4:"INT(M(4))
950 PRINT "-----------------------------
---"
960 PRINT "TOTAL OUTPUT IS"INT(M(5))
970 PRINT "-----------------------------
---"
980 PRINT "EFFICIENCY LEVEL IS"INT(EF)"%
"
990 RETURN
1000 REM **************************
1010 REM GET REPLIES
1020 A$=INKEY$
1030 IF A$<>"Y" AND A$<>"N" THEN 1020
1040 PRINT TAB(22);A$
1050 FOR J=1 TO 500:NEXT J
1060 RETURN
1070 REM **************************
1080 REM INCREASE OUTPUT?
1090 IF M(4)=0 THEN X=15:GOTO 1110
1100 X=18
```

```
1110 PRINT "IT WILL COST $M"X" TO EXPAND
"
1120 PRINT TAB(8);"OUTPUT BY 1%"
1130 PRINT "-------------------------
----"
1140 PRINT "HOW MANY % DO YOU WISH TO RA
ISE OUTPUT?"
1150 INPUT EP:IF EP<0 OR EP>100 THEN 115
0
1160 M(5)=0
1170 FOR T=1 TO 4
1180 M(T)=M(T)+M(T)*EP/100
1190 M(5)=M(5)+M(T)
1200 NEXT T
1210 TP=TP-EP*X
1220 FOR T=1 TO 500:NEXT T
1230 GOTO 170
1240 REM **************************
1250 REM SALE OF FACTORY FOUR
1260 PRINT "FACTORY 4 IS VALUED FOR SALE
 AT $M104"
1270 PRINT "YOU CAN'T REBUY IT LATER IF
          YOU SELL IT..."
1280 PRINT "DO YOU WANT TO SELL (Y/N)?"
1290 GOSUB 1010
1300 IF A$="N" THEN 210
1310 TP=TP+104
1320 SF=1
1330 M(5)=M(1)+M(2)+M(3)
1340 M(4)=0
1350 GOTO 170
1360 REM **************************
1370 REM CHECK ON LOSSES
1380 IF MP>0 THEN SA=0:GOTO 1480
1390 SA=SA+1
1400 IF SA>11 THEN 1420
1410 GOTO 1480
1420 CLS:PRINT
1430 PRINT "YOU JUST MADE YOUR TWELFTH M
ONTHLY"
1440 PRINT "LOSS IN A ROW.............
..."
```

```
1450 PRINT TAB(6);"YOUR EMPLOYMENT"
1460 PRINT TAB(6);"IS HEREBY TERMINATED!
!"
1470 END
1480 IF TP>=-250 THEN 1530
1490 CLS:PRINT
1500 PRINT "UNDER YOUR MANAGEMENT, THE C
OMPANY HAS"
1510 PRINT "LOST MORE THAN $M250!"
1520 GOTO 1450
1530 IF TP>200 THEN 1570
1540 RETURN
1550 REM **************************
1560 REM SWEET SWEET SUCCESS!!!
1570 CLS:PRINT
1580 PRINT "WELL DONE! THE COMPANY HAS M
ADE MORE"
1590 PRINT "   THAN $M200.    YOU'VE BEE
N MADE"
1600 PRINT "          A MEMBER OF THE BOAR
D"
1610 FOR T=1 TO 2000:NEXT T
1620 END
1630 REM **************************
1640 REM INITIALIZATION
1650 CLS
1660 RANDOMIZE VAL(RIGHT$(TIME$,2))
1670 DIM M(5),Y(5)
1680 NE=12000:REM STARTING NO EMPLOYEES
1690 AW=22995:REM STARTING AVERAGE WAGE
1700 AC=11100:REM COST PRICE/VEHICLE
1710 AS=12000:REM SELLING PRICE
1720 MI=50*1000:MC=10100
1730 Y(3)=12500
1740 MS=25:EF=77:FA=160:SF=0:MT=0
1750 FOR J=1 TO 5
1760 READ M(J)
1770 NEXT J
1780 RETURN
1790 DATA 8900,3250,2500,1625,16275
```

Chapter Seventeen
Life at the Super Bowl

It all started back in 1823, at a very upper class public school in England. Rugby School student, William Webb Ellis, was playing soccer one day, when he picked up the ball, tucked it under his arm, and ran frantically across his opponent's goal. This action was the moment in which the game which became rugby football was born. From rugby football came rugby league, and Gridiron.

Some students from Harvard in 1874 saw the game, as it had then evolved, underway at Montreal University, and liked what they saw. They took some of the Montreal rules and used them to modify a soccer-like game which was already gaining favour in the States.

The game took off, but was so rough, and caused so many injuries, that at the turn of the century many colleges called for it to be banned. However, Teddy Roosevelt, who liked the game (and most other rugged sports), suggested that instead of banning it, the game should be made simpler. A committee, formed of leading coaches, managers and players, met in 1906 to follow Roosevelt's suggestion, and from this committee comes the Gridiron we know today.

In our program, you can either play against the computer team (the 'Silicon Cowboys') or against another human being. In the game shown here, I decided to play against the computer:

```
ONE PLAYER OR TWO
? 1
AND THE NAME OF THE VISITING TEAM?
? HARTNELL'S RAIDERS
```

In this simulation, you have the option of throwing, carrying or punting (including field goal attempts) the ball. The program

allows for four 15 minute quarters, and tests your decision-making skills and reflexes.

```
THERE ARE 60 MINUTES TO GO
        SILICON COWBOYS TO KICK OFF
YOU ARE ON YOUR OWN 35 YARD LINE
SILICON COWBOYS HAVE...
KICKED 1 YARDS
KICKED 2 YARDS
KICKED 3 YARDS

                    KICKED 52 YARDS
                     KICKED 53 YARDS
                     KICKED 54 YARDS
                     KICKED 55 YARDS
THE BALL IS CAUGHT!
AND RETURNED 1 YARDS
AND RETURNED 2 YARDS
AND RETURNED 3 YARDS

     AND RETURNED 27 YARDS
     AND RETURNED 28 YARDS
     AND RETURNED 29 YARDS
--------------------------------
THE BALL IS DOWN ON
HARTNELL'S RAIDERS'S 39 YARD LINE
      > PRESS ANY KEY <


SILICO 0 HARTNE 0
 60 MINUTES TO GO
--------------------------------
HARTNELL'S RAIDERS IN POSSESSION
     0 DOWN
     10 YARDS TO GO
--------------------------------
START AT 39 YARD LINE
NOW ON 39 YARD LINE
 61 YARDS TO TOUCHDOWN
```

```
--------------------------------
ON THIS PLAY HARTNELL'S RAIDERS CAN
EITHER 1 - THROW
       2 - CARRY
    OR 3 - PUNT
```

Once you've decided on your play, the computer will begin a count from one to eleven. You have to get rid of the ball during that count. If you don't do so in time, you'll be 'sacked' and lose yardage or possession.

```
HARTNELL'S RAIDERS, YOUR QUARTERBACK HAS

        GOT THE BALL
--------------------------------
WAIT FOR THE COUNT, HARTNELL'S RAIDERS,
        THEN HIT ANY KEY...
 1
  2
   3
    4
     5
      6
--------------------------------
NICE PUNT...
YOU'VE KICKED 24 YARDS
--------------------------------
```

The simulation tempts you to wait as long as possible before delivering the ball, because the longer you wait, the further the ball will travel. For example, if you decide to throw the ball, and you stop the count on two, the ball will only travel 1Ø yards, whereas it would have gone 4Ø yards if you'd stopped the count on 1Ø.

```
THE BALL IS CAUGHT!
AND RETURNED 1 YARDS
AND RETURNED 2 YARDS
AND RETURNED 3 YARDS

AND RETURNED 17 YARDS
 AND RETURNED 18 YARDS
 AND RETURNED 19 YARDS
 AND RETURNED 20 YARDS
```

```
----------------------------------
THE BALL IS DOWN ON
SILICON COWBOYS'S 58 YARD LINE


SILICO 0 HARTNE 0
 59.7 MINUTES TO GO
----------------------------------
SILICON COWBOYS IN POSSESSION
     0 DOWN
     10 YARDS TO GO
----------------------------------
START AT 58 YARD LINE
NOW ON 58 YARD LINE
 42 YARDS TO TOUCHDOWN
----------------------------------
ON THIS PLAY SILICON COWBOYS CAN
EITHER 1 - THROW
       2 - CARRY
    OR 3 - PUNT
```

As you can see, the fact that you've thrown a given distance does not mean the play will automatically be complete. The chances of a play being complete are linked to the distance thrown.

```
  TOUCHDOWN!!!
    TOUCHDOWN!!!
      TOUCHDOWN!!!
        TOUCHDOWN!!!
          TOUCHDOWN!!!
SILICON COWBOYS 6
HARTNELL'S RAIDERS 0
TO PLAY FOR EXTRA POINT
     > PRESS ANY KEY <
```

There is a lot of action in this simulation, and you're sure to enjoy 'dressing it up' some more. Here are a few additional scenes of the program in action:

```
      AND RETURNED 27 YARDS
       AND RETURNED 28 YARDS
       AND RETURNED 29 YARDS
       AND RETURNED 30 YARDS
       AND RETURNED 31 YARDS
--------------------------------
THE BALL IS DOWN ON
HARTNELL'S RAIDERS'S 44 YARD LINE
      > PRESS ANY KEY <
                          OK
 PERIOD OVER
    PERIOD OVER
       PERIOD OVER
          PERIOD OVER
             PERIOD OVER
                PERIOD OVER
                   PERIOD OVER
                      PERIOD OVER
                         PERIOD OVER
                            PERIOD OVER
                               PERIOD OVER
SILICON COWBOYS 27
HARTNELL'S RAIDERS 6
      > PRESS ANY KEY <


                  PERIOD OVER
                     PERIOD OVER
                        PERIOD OVER
                           PERIOD OVER
                              PERIOD OVER
                                 PERIOD OVER
SILICON COWBOYS 27
HARTNELL'S RAIDERS 6
      > PRESS ANY KEY <
                          OK
  15 MINUTES TO GO

--------------------------------
HARTNELL'S RAIDERS IN POSSESSION
      0 DOWN
      10 YARDS TO GO
```

```
--------------------------------
START AT 44 YARD LINE
NOW ON 44 YARD LINE
 56 YARDS TO TOUCHDOWN
--------------------------------
ON THIS PLAY HARTNELL'S RAIDERS CAN
EITHER 1 - THROW
       2 - CARRY
    OR 3 - PUNT

    3
     4
      5
       6
        7
--------------------------------
GREAT RUNNING BY THE OPPOSITION HAS
CAUSED YOU TO LOSE 3 YARDS
      > PRESS ANY KEY <
                      OK
 GAME OVER
    GAME OVER
       GAME OVER
          GAME OVER
             GAME OVER
                GAME OVER
                   GAME OVER
                      GAME OVER
                         GAME OVER
                            GAME OVER
                               GAME OVER
SILICON COWBOYS 40
HARTNELL'S RAIDERS 6
```

The listing of *Gridiron* is in the next chapter:

Chapter Eighteen
The Gridiron Listing

When you want to become the whole Green Bay Packers team, the New York Jets or the Miami Dolphins, enter and run this listing:

```
10 REM GRIDIRON
20 CLS
30 RANDOMIZE VAL(RIGHT$(TIME$,2))
40 GOSUB 70
50 GOTO 220
60 REM ************************
70 FOR X=1 TO 1500:NEXT X
80 RETURN
90 REM ************************
100 PRINT A$;SA
110 PRINT B$;SB
120 RETURN
130 REM ************************
140 IF Z$=A$ THEN Z$=B$:RETURN
150 Z$=A$:RETURN
160 REM ************************
170 IF INKEY$<>"" THEN 170
180 PRINT "      > PRESS ANY KEY <"
190 IF INKEY$="" THEN 190
200 PRINT TAB(20);"OK":RETURN
210 REM ************************
220 REM INITIALIZE
230 DEF FNA(X)=INT(RND(1)*X)+1
240 PRINT "ONE PLAYER OR TWO"
250 INPUT X
260 IF X<1 OR X>2 THEN 250
270 IF X=1 THEN VC=1:A$="SILICON COWBOYS
":GOTO 300
280 PRINT "WHAT IS THE NAME OF THE HOME
TEAM?"
```

```
290 INPUT A$:IF A$="" THEN 290
300 PRINT "AND THE NAME OF THE VISITING
TEAM?"
310 INPUT B$:IF B$="" THEN 310
320 Z$=A$:NU=35
330 CLS
340 PRINT "THERE ARE"INT(10*(60-(W/4)))/
10"MINUTES TO GO"
350 PRINT TAB(8);Z$" TO KICK OFF"
360 PRINT "YOU ARE ON YOUR OWN"NU"YARD L
INE"
370 IF VC=1 AND Z$=A$ THEN GOSUB 70:GOTO
 400
380 PRINT "TO KICK OFF..."
390 GOSUB 170
400 A=FNA(20)+40
410 PRINT Z$;" HAVE..."
420 FOR X=1 TO A
430 PRINT TAB(X/3);"KICKED"X"YARDS"
440 NEXT X
450 NU=NU+X
460 GOSUB 70
470 PRINT "THE BALL IS CAUGHT!"
480 GOSUB 70
490 A=FNA(30)+10
500 FOR X=1 TO A
510 PRINT TAB(X/5);"AND RETURNED"X"YARDS
"
520 NEXT X
530 NU=ABS(100-NU+X)
540 GOSUB 140
550 PRINT "-------------------------------
----"
560 PRINT "THE BALL IS DOWN ON"
570 PRINT Z$"'S"NU"YARD LINE"
580 IF Z$=A$ AND VC=1 THEN GOSUB 70:GOTO
 600
590 GOSUB 170
600 TG=10:D=0:SL=NU
610 IF W=60 OR W=180 THEN 2010
620 IF W=120 THEN 2070
```

```
630 IF W=240 THEN 2140
640 CLS
650 PRINT LEFT$(A$,6);SA;LEFT$(B$,6);SB
660 PRINT INT(10*(60-(W/4)))/10"MINUTES
TO GO"
670 GOSUB 70
680 PRINT "-------------------------
----"
690 PRINT Z$" IN POSSESSION"
700 PRINT TAB(4);D"DOWN"
710 PRINT TAB(4);TG"YARDS TO GO"
720 PRINT "-------------------------
----"
730 PRINT "START AT"SL"YARD LINE"
740 PRINT "NOW ON"NU"YARD LINE"
750 PRINT 100-NU"YARDS TO TOUCHDOWN"
760 PRINT "-------------------------
----"
770 PRINT "ON THIS PLAY ";
780 IF Z$=A$ THEN PRINT A$;" CAN":GOTO 8
00
790 PRINT B$;" CAN"
800 PRINT "EITHER 1 - THROW"
810 PRINT "       2 - CARRY"
820 PRINT "    OR 3 - PUNT"
830 P=0
840 IF Z$=A$ AND VC=1 AND D<3 THEN P=2:G
OTO 900
850 IF Z$=A$ AND VC=1 AND TG<7 THEN P=2:
GOTO 900
860 IF Z$=A$ AND VC=1 AND (100-NU)<31 TH
EN P=3:GOTO 900
870 IF Z$=A$ AND VC=1 THEN P=1:GOTO 900
880 K$=INKEY$:IF K$<"1" OR K$>"3" THEN 8
80
890 P=VAL(K$):PRINT TAB(10);"OK"P
900 GOSUB 70
910 W=W+1
920 CLS
930 PRINT Z$;", YOUR QUARTERBACK HAS"
940 PRINT TAB(8);"GOT THE BALL"
```

```
950 PRINT "----------------------------
----"
960 PRINT "WAIT FOR THE COUNT, ";Z$;","
970 PRINT TAB(8);"THEN HIT ANY KEY..."
980 IF INKEY$<>"" THEN 980
990 GOSUB 70
1000 GOSUB 2200
1010 IF E=11 THEN 2340
1020 PRINT "----------------------------
-----"
1030 ON P GOTO 1050,1310,1590
1040 REM **************************
1050 PRINT "YOU'VE THROWN"E*5"YARDS"
1060 PRINT TAB(4);"AND THE PLAY IS..."
1070 A=FNA(8)
1080 IF A=1 THEN 1520
1090 A=FNA(E+1)
1100 IF A=1 THEN PRINT TAB(20);"...COMPL
ETE":GOTO 1220
1110 PRINT TAB(20);"...INCOMPLETE":D=D+1
1120 GOSUB 170
1130 PRINT "----------------------------
------"
1140 IF D>3 THEN 1160
1150 GOTO 610
1160 PRINT "THAT WAS YOUR 4TH DOWN"
1170 PRINT "AND YOU'VE LOST POSSESSION!!
"
1180 D=0:TG=10:NU=ABS(100-NU):SL=NU
1190 GOSUB 70
1200 GOSUB 140
1210 GOTO 610
1220 GOSUB 170
1230 NU=NU+(E*5):TG=TG-(E*5)
1240 IF NU>100 THEN 1800
1250 IF TG<1 THEN 1280
1260 D=D+1:IF D>3 THEN 1160
1270 GOTO 610
1280 D=0:TG=10:SL=NU
1290 GOTO 610
```

```
1300 REM **************************
1310 A=FNA(15)
1320 IF A=1 THEN 1510
1330 E=A-5
1340 IF E<0 THEN 1440
1350 IF E=0 THEN E=1:GOTO 1370
1360 PRINT "GOOD SNAP, PASS AND RUN"
1370 PRINT "YOU'VE GAINED"E"YARDS"
1380 GOSUB 170
1390 TG=TG-E:NU=ABS(NU+E):D=D+1
1400 IF NU>100 THEN 1800
1410 IF TG<1 THEN 1280
1420 IF D>3 THEN 1160
1430 GOTO 610
1440 PRINT "GREAT RUNNING BY THE OPPOSIT
ION HAS"
1450 PRINT "CAUSED YOU TO LOSE"ABS(E)"YA
RDS"
1460 TG=TG-E:NU=NU+E:D=D+1
1470 GOSUB 170
1480 IF D>3 THEN 1160
1490 GOTO 610
1500 REM **************************
1510 PRINT "BAD SNAP...YOU'VE"
1520 PRINT "FUMBLED...AND"
1530 PRINT "YOU'VE LOST POSSESSION..."
1540 NU=100-NU:D=0:TG=10:SL=NU
1550 REM **************************
1560 GOSUB 170
1570 GOTO 460
1580 REM **************************
1590 PRINT "NICE PUNT..."
1600 PRINT "YOU'VE KICKED"E*4"YARDS"
1610 NU=NU+E*4
1620 IF NU>100 THEN 1650
1630 PRINT "-----------------------------
-----"
1640 GOTO 460
1650 A=FNA(3)
1660 IF A>1 THEN 1740
1670 PRINT "BUT YOU'VE MISSED THE GOAL!!
"
```

```
1680 IF NU-E*4<80 THEN NU=ABS[100-[NU-E*
4]]:GOTO 1700
1690 NU=20
1700 D=0:TG=10:SL=NU
1710 GOSUB 140
1720 GOSUB 170
1730 GOTO 610
1740 PRINT ".....AND SCORED!"
1750 IF Z$=B$ THEN SB=SB+3:GOTO 1770
1760 SA=SA+3
1770 GOSUB 100
1780 GOSUB 170
1790 NU=35:GOTO 330
1800 CLS
1810 FOR X=1 TO 5
1820 PRINT TAB[X*2];"TOUCHDOWN!!!"
1830 NEXT X
1840 IF Z$=A$ THEN SA=SA+6:GOTO 1860
1850 SB=SB+6
1860 GOSUB 100
1870 PRINT "TO PLAY FOR EXTRA POINT"
1880 GOSUB 170
1890 PRINT "-----------------------------
-----"
1900 PRINT "THE BALL IS SNAPPED...PREPAR
E TO KICK!"
1910 GOSUB 70
1920 GOSUB 2200
1930 IF E>9 THEN PRINT "YOU MISSED":NU=2
0:GOTO 1970
1940 PRINT "YOU SCORED...":NU=35
1950 IF Z$=A$ THEN SA=SA+1:GOTO 1980
1960 SB=SB+1:GOTO 1980
1970 GOSUB 140
1980 GOSUB 100
1990 GOSUB 170
2000 GOTO 330
2010 FOR X=1 TO 10
2020 PRINT TAB[2*X];"PERIOD OVER"
2030 NEXT X
2040 GOSUB 100
```

```
2050 GOSUB 170
2060 GOTO 660
2070 FOR X=1 TO 10
2080 PRINT TAB(2*X);"HALF TIME"
2090 NEXT X
2100 GOSUB 100
2110 Z$=B$
2120 GOSUB 170
2130 NU=35:W=W+2:GOTO 330
2140 FOR X=1 TO 10
2150 PRINT TAB(2*X);"GAME OVER"
2160 NEXT X
2170 GOSUB 100
2180 END
2190 REM ************************
2200 E=0:X=10
2210 IF Z$=A$ AND VC=1 THEN PRINT "THIS
ONE FOR ";A$:GOTO 2290
2220 E=E+1:X=X-1
2230 PRINT TAB(E);E
2240 FOR Y=1 TO X*1.5
2250 IF INKEY$<>"" THEN Y=X*1.5+1:RETURN

2260 NEXT Y
2270 IF E=11 THEN RETURN
2280 GOTO 2220
2290 FOR E=1 TO FNA(7)+2
2300 FOR J=1 TO 60:NEXT J
2310 PRINT TAB(E);E
2320 NEXT E
2330 RETURN
2340 PRINT "TOO LATE!"
2350 PRINT "YOU'VE BEEN SACKED!"
2360 E=FNA(4)
2370 IF E=3 THEN 2430
2380 PRINT "AND LOST FIVE YARDS!"
2390 TG=TG+5:D=D+1:NU=NU-5
2400 GOSUB 170
2410 IF D>3 THEN 1160
2420 GOTO 610
2430 PRINT "AND LOST POSSESSION!"
```

```
2440  D=0:NU=ABS(100-NU+5):SL=NU:TG=10
2450  GOSUB 170
2460  GOSUB 140
2470  GOTO 610
```

Chapter Nineteen
The Grand Slam

In this sports simulation, you have a chance to play a three-set game of tennis against your computer. If you prefer, you can play against another human being, with the simulation reporting on, and moderating, the action.

Tennis got underway in England in the middle of the last century. By 1877 it was such a popular game, that a national championship at Wimbledon was staged. There is a maximum of five sets in a match for men, and three for women, with a minimum of six games per set. You need to lead by two in order to win a set.

The simulation calls on your decision-making skill and tests your reflexes. Your first decision is the speed of your serve. If you elect to play against the computer, it will always serve first, in order to allow you to see it in action. You'll learn more about the simulation from watching the computer serve than from reading this introduction.

The simulation is weighted so that fast serves have a better chance of an ace than do slow serves, but there is also a greater chance of a fault. Slow serves will rarely give an ace, but as well are unlikely to fault.

After service, the simulation calls on your reflexes. The screen clears and indicates whose court the ball is in. After a brief pause, the computer will begin a countdown to –1. To hit the ball, you need to wait until the count reaches zero, and then press any key.

If you 'swing' too soon or too late, you'll miss the ball and the computer will tell you whether the ball was in or out of play. Note that you can choose to leave a ball, if you think it is going to go out. If you 'hit' the ball, by striking a key just as the count reaches zero,

the screen will clear and show the ball in the other player's court. This process continues until one player or the other misses the ball.

The program begins by asking you whether there will be one human player or two. If there is only one, it will ask for the player's name with the prompt NAME OF SECOND PLAYER?:

```
ONE HUMAN PLAYER OR TWO? 1
NAME OF SECOND PLAYER? TIMOTY
```

This simulation needs names which are six letters long. If you enter a longer name, the computer will cut it down to size. The computer player's name, as you'll see in a moment, is 'Bjornx'. The match begins as Bjornx serves:

```
BJORNX SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

        1   > IT'S A FAST SERVE...
```

Bjornx elects for a fast serve, and the mere human must attempt to return it:

```
TIMOTY, THE BALL IS
IN YOUR COURT
----------------------
HIT ANY KEY, WHEN YOU SEE THE ZERO,
      TO RETURN THE BALL...
                5
                   4
                      3
                         2
                            1
YOU MISSED THE BALL, AND...
    IT WAS IN...BAD MISTAKE
```

Unfortunately, Timoty is not playing with his usual fine form. The score is shown, before Bjornx serves again:

```
-----------------------
              SET SET SET
-----------------------
                1   2   3  GAME
BJORNX          0   0   0  30
TIMOTY          0   0   0  0
-----------------------
BJORNX SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

1  > IT'S A FAST SERVE...
```

The scoring system of our simulation follows the real game, with a minimum of four points needed to win a game: 15, 30, 40 and game. If both sides score 4Ø, 'deuce' is declared, and the first player to score a point is said to have the 'advantage'. A further point is needed to win. If the opponent gets a point, instead of the player with the 'advantage', the score returns to 'deuce'. Of course, you do not have to worry about this scoring system, as the computer obligingly keeps track of everything for you.

Here are a few more 'snapshots' of the game in progress:

```
GAME TO BJORNX


-----------------------
              SET SET SET
-----------------------
                1   2   3  GAME
BJORNX          1   0   0  0
TIMOTY          0   0   0  0
-----------------------
TIMOTY SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

1  > IT'S A FAST SERVE...
```

```
                ....OUT....
          ...SECOND SERVE...
TIMOTY SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

       2   > IT'S A SLOW SERVE...

-----------------------
              SET SET SET
-----------------------
                1   2   3   GAME
BJORNX          6   0   0   0
TIMOTY          1   0   0   40
-----------------------
TIMOTY SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

       2   > IT'S A SLOW SERVE...

-----------------------
              SET SET SET
-----------------------
                1   2   3   GAME
BJORNX          6   0   0   15
TIMOTY          1   1   0   0
-----------------------
BJORNX SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

       1   > IT'S A FAST SERVE...
```

It appears that the human race is no match for the mighty computer:

```
-----------------------
              SET SET SET
-----------------------
               1   2   3   GAME
BJORNX         6   4   0   30
TIMOTY         1   2   0   40
-----------------------
TIMOTY SERVING
DO YOU WANT TO SERVE 1 - FAST
                     OR 2 - SLOW

      2   > IT'S A SLOW SERVE...

-----------------------
              SET SET SET
-----------------------
               1   2   3   GAME
BJORNX         6   4   0   {DEUCE
TIMOTY         1   2   0   {DEUCE
-----------------------

-----------------------
              SET SET SET
-----------------------
               1   2   3   GAME
BJORNX         6   6   6   0
TIMOTY         1   4   4   0
-----------------------
```

Chapter Twenty
The Tennis Listing

When you're ready to take on Bjornx for a set or two, enter and run the following listing:

```
10 REM TENNIS
20 CLS
30 RANDOMIZE VAL(RIGHT$(TIME$,2))
40 AA=0:BB=0:T=0:KA=0
50 XA=0:YA=0:ZA=0
60 XB=0:YB=0:ZB=0
70 DEF FNA(X)=INT(RND(1)*X)+1
80 INPUT "ONE HUMAN PLAYER OR TWO";A
90 IF A<1 OR A>2 THEN 80
100 IF A=1 THEN A$="BJORNX":VC=1
110 IF VC=1 THEN 160
120 PRINT "PLEASE ENTER A SIX-LETTER NAM
E"
130 INPUT "NAME OF FIRST PLAYER";A$
140 IF LEN(A$)<6 THEN A$=A$+" ":GOTO 140

150 A$=LEFT$(A$,6)
160 INPUT "NAME OF SECOND PLAYER";B$
170 IF LEN(B$)<6 THEN B$=B$+" ":GOTO 170

180 B$=LEFT$(B$,6)
190 S=1:AA=1:BB=1
200 CLS
210 P$=A$:R$=B$
220 REM ****************************
230 IF P$=A$ THEN R$=B$
240 IF P$=B$ THEN R$=A$
250 PRINT P$;" SERVING"
260 PRINT "DO YOU WANT TO SERVE 1 - FAST
"
```

```
270 PRINT "                              OR 2 - SLOW
"
280 IF P$=A$ AND VC=1 AND SC=0 THEN KB=1
:GOSUB 1720:GOTO 330
290 IF P$=A$ AND VC=1 AND SC=1 THEN KB=2
:GOSUB 1720:GOTO 330
300 K$=INKEY$
310 IF K$<"1" OR K$>"2" THEN 300
320 KB=VAL(K$)
330 PRINT:PRINT TAB(6);KB;TAB(10);"> IT'
S A ";
340 IF KB=1 THEN PRINT "FAST";
350 IF KB=2 THEN PRINT "SLOW";
360 PRINT " SERVE..."
370 GOSUB 1720
380 IF KB=1 THEN EB=FNA(3):GOTO 400
390 EB=FNA(8)
400 IF EB=1 THEN 450
410 IF EB=3 AND SC=0 THEN 520
420 IF EB=3 AND SC=1 THEN 590
430 GOTO 670
440 REM ******************************
450 CLS:PRINT
460 PRINT TAB(8);"....ACE...."
470 GOSUB 1720
480 SC=0
490 IF P$=A$ THEN 1140
500 GOTO 1150
510 REM ******************************
520 CLS:PRINT
530 PRINT TAB(12);"....OUT...."
540 PRINT TAB(8);"...SECOND SERVE..."
550 GOSUB 1720
560 SC=1
570 GOTO 230
580 REM ******************************
590 CLS:PRINT
600 PRINT TAB(12);"....OUT...."
610 PRINT TAB(9);"....DOUBLE FAULT..."
620 GOSUB 1720
630 SC=0
```

```
640 IF P$=A$ THEN 1150
650 GOTO 1140
660 REM ****************************
670 SC=0
680 CLS:PRINT
690 IF INKEY$<>"" THEN 690
700 PRINT R$;", THE BALL IS":PRINT "IN Y
OUR COURT"
710 PRINT "----------------------"
720 IF R$=A$ AND VC=1 THEN 750
730 PRINT "HIT ANY KEY, WHEN YOU SEE THE
 ZERO,              TO RETURN THE BALL..."
740 IF INKEY$<>"" THEN 740
750 X=4*FNA(3):Y=X
760 GOSUB 1720
770 E=5
780 PRINT TAB(2*(11-E));E
790 Y=Y-1
800 S$=INKEY$
810 IF S$<>"" AND E=0 THEN 890
820 IF S$<>"" THEN 990
830 IF Y>0 THEN 790
840 E=E-1:Y=X
850 IF E<-1 THEN 990
860 IF E=-1 AND R$=A$ AND VC=1 THEN 890
870 GOTO 780
880 IF KB=1 THEN EA=FNA(2):GOTO 1000
890 EA=FNA(4)
900 IF E=0 AND R$=A$ AND VC=1 THEN EA=FN
A(8)
910 IF EA=1 THEN 940
920 IF R$=A$ THEN R$=B$:GOTO 670
930 R$=A$:GOTO 670
940 PRINT R$;", YOU'VE HIT THE BALL"
950 PRINT TAB(8);"OUT OF PLAY..."
960 GOSUB 1720
970 IF R$=A$ THEN R$=B$:GOTO 1150
980 GOTO 1140
990 EA=FNA(4)
1000 IF EA=1 THEN 1070
1010 PRINT "YOU MISSED THE BALL, AND..."
```

```
1020 GOSUB 1720
1030 PRINT "    IT WAS IN...BAD MISTAKE"
1040 GOSUB 1720
1050 IF R$=A$ THEN R$=B$:GOTO 1150
1060 GOTO 1140
1070 PRINT "YOU MISSED THE BALL AND..."
1080 GOSUB 1720
1090 PRINT "    IT WAS OUT..WELL LEFT"
1100 GOSUB 1720
1110 IF R$=A$ THEN R$=B$:GOTO 1140
1120 GOTO 1150
1130 REM ***************************
1140 AA=AA+1:GOTO 1160
1150 BB=BB+1
1160 IF AA<5 AND BB<5 THEN 1230
1170 IF [BB>4 AND AA<4] OR [BB>4 AND BB-
AA>1] THEN AA=1:BB=1:GOTO 1500
1180 IF [AA>4 AND BB<4] OR [AA>4 AND AA-
BB>1] THEN AA=1:BB=1:GOTO 1440
1190 IF AA>4 AND AA>BB THEN C$="ADV":D$=
"---":GOTO 1320
1200 IF BB>4 AND BB>AA THEN D$="ADV":C$=
"---":GOTO 1320
1210 C$="{DEUCE":D$="{DEUCE":GOTO 1320
1220 REM ***************************
1230 RESTORE
1240 FOR D=1 TO AA
1250 READ C$
1260 NEXT D
1270 RESTORE
1280 FOR D=1 TO BB
1290 READ D$
1300 NEXT D
1310 REM ***************************
1320 CLS
1330 PRINT "-------------------------"
1340 PRINT "              SET SET SET"
1350 PRINT "-------------------------"
1360 PRINT "                1   2   3  GAME"
1370 PRINT A$;"      ";XA;" ";YA;" ";ZA;"
 ";C$
```

```
1380 PRINT B$;"      ";XB;" ";YB;" ";ZB;"
 ";D$
1390 PRINT "----------------------"
1400 GOSUB 1720
1410 IF T<>1 THEN 230
1420 END
1430 REM ************************
1440 CLS
1450 PRINT "GAME TO ";A$
1460 GOSUB 1720
1470 IF S=1 THEN XA=XA+1:C$="O":D$="O":G
OTO 1560
1480 IF S=2 THEN YA=YA+1:C$="O":D$="O":G
OTO 1580
1490 IF S=3 THEN ZA=ZA+1:C$="O":D$="O":G
OTO 1600
1500 CLS
1510 PRINT "GAME TO ";B$
1520 GOSUB 1720
1530 IF S=1 THEN XB=XB+1:C$="O":D$="O":G
OTO 1560
1540 IF S=2 THEN YB=YB+1:C$="O":D$="O":G
OTO 1580
1550 IF S=3 THEN ZB=ZB+1:C$="O":D$="O":G
OTO 1600
1560 IF (XA>5 AND XB<5) OR (XA<5 AND XB>
5) THEN 1630
1570 IF (XA>5 AND XA-XB>1) OR (XB>5 AND
XB-XA>1) THEN 1630
1580 IF (YA>5 AND YB<5) OR (YA<5 AND YB>
5) THEN 1630
1590 IF (YA>5 AND YA-YB<1) OR (YB>5 AND
YB-YA>1) THEN 1630
1600 IF (ZA>5 AND ZB<5) OR (ZA<5 AND ZB>
5) THEN 1680
1610 IF (ZA>5 AND ZA-ZB>1) OR (ZB>5 AND
ZB-ZA>1) THEN 1680
1620 GOTO 1640
1630 S=S+1
1640 AA=1:BB=1
1650 IF P$=A$ THEN R$=A$:P$=B$:GOTO 1320
```

```
1660 P$=A$:R$=B$:GOTO 1320
1670 REM ********************************
1680 T=1
1690 GOTO 1320
1700 REM ********************************
1710 REM DELAY
1720 FOR M=1 TO 1000:NEXT M
1730 RETURN
1740 DATA "0","15","30","40"
```

Chapter Twenty-One
Driving A Racing Car

Now you have the chance to drive a simulated racing car around your choice of four of the world's greatest Grand Prix circuits. You, too, can be a hero on the British course known as 'The Whale'; at Germany's 'The Key'; on the Italian track called 'The Shoe'; or at Monaco on the 3100 metre track.

The program begins by getting your name, and wishing you luck, before asking you which track you wish to try, and how many laps the race will be over:

```
WHAT IS YOUR NAME, DRIVER? HARTNELL
    OK, GOOD LUCK, HARTNELL
        OK, GOOD LUCK, HARTNELL
            OK, GOOD LUCK, HARTNELL
**********************************************
WHICH RACE DO YOU WANT TO TAKE PART IN:

        BRITISH GRAND PRIX   2650MT   :1
        GERMAN  GRAND PRIX   1700MT   :2
        ITALIAN GRAND PRIX   2200MT   :3
        MONACO  GRAND PRIX   3100MT   :4

          ENTER A NUMBER [1 TO 4]
**********************************************
          OK, THE BRITISH RACE
**********************************************

OVER HOW MANY LAPS?
? 1
```

The aim of the simulation, naturally enough, is to complete the race as quickly as possible, without crashing and without overheating either your engine or brakes.

You start the race in position six. On the British circuit, you start the simulation at 14Ø kilometres an hour, just under the speed recommended for the approaching corner. To increase your speed, you hit the "M" key. The "Z" will slow you down, and the space bar enables you to maintain the same speed. A few seconds after the simulation begins, the screen looks like this:

```
ENGINE TEMPERATURE 100 C.   (MAX. 200)
BRAKE TEMPERATURE: 10 C.   (MAX. 500)
 DISTANCE COVERED: 0 METERS
                 : 0 LAPS
YOU'RE IN POSITION 6
-----------------------------------
    CURRENT SPEED: 140 KPH
                 : 77.7 METERS PER MOVE
-----------------------------------
CORNER APPROACHING 800 METERS
RECOMMENDED SPEED: 150 KPH
-----------------------------------
```

The simulation run began with the first corner 8ØØ metres away. This has now shrunk to just 57 metres, and I've increased my speed to 148 kph, just below the recommended speed for taking that corner of 15Ø kph. The engine has warmed up slightly (but still a long, long way away from the maximum of 2ØØ), and the temperature of the brakes has gone up from 1Ø degrees to nearly 55.

```
ENGINE TEMPERATURE 104.6 C.   (MAX. 200)
BRAKE TEMPERATURE: 54.7 C.   (MAX. 500)
 DISTANCE COVERED: 742.2 METERS
                 : .28 LAPS
YOU'RE IN POSITION 6
-----------------------------------
    CURRENT SPEED: 148 KPH
                 : 82.2 METERS PER MOVE
-----------------------------------
CORNER APPROACHING 57 METERS
RECOMMENDED SPEED: 150 KPH
-----------------------------------
```

I safely negotiate that corner, but find that I have, perhaps, been a bit too cautious, as with just under half of the one lap race completed, I am now in position seven (after starting in position six):

```
ENGINE TEMPERATURE 87.1 C.  (MAX. 200)
BRAKE TEMPERATURE: 157.3 C.  (MAX. 500)
 DISTANCE COVERED: 1102 METERS
                 : .41 LAPS
YOU'RE IN POSITION 7
-------------------------------
    CURRENT SPEED: 112.3 KPH
                 : 62.3 METERS PER MOVE
-------------------------------
CORNER APPROACHING 97 METERS
RECOMMENDED SPEED: 90 KPH
-------------------------------
```

I am just 97 metres from the next corner, where the recommended cornering speed is 90 kph. I'm doing just over 112 kph. Frantically, I slam on the brakes, hitting hell out of the "Z" key. But it is too late:

```
YOU CORNERED AT 104.8 KPH
AND THE MAXIMUM SPEED WAS JUST 90
YOU SPIN OFF THE TRACK...
..........AND CRASH!!!!!
-------------------------------
YOU ONLY COMPLETED 1218.4 METERS,
OR .45 LAPS AND AT THAT
STAGE WERE IN POSITION 7
-------------------------------
YOUR AVERAGE SPEED WAS 137.07 KPH
YOU WERE 2 TH FASTEST ON STRAIGHTS,
AND 12 TH FASTEST ON CORNERS.

PRESS 'S' FOR SAME RACE, 'N' FOR NEW
 RACE, 'E' TO END
```

I was in seventh position when the race ended. The final screen tells me my average speed was a shade over 137 kph, I was second fastest on the straights, and only twelfth fastest on the corners.

As you can see, you are given the choice of running the same race again (by pressing "S"), choosing a new race ("N") or ending the simulation ("E").

I press "N" and decide on the German circuit:

```
**********************************
WHICH RACE DO YOU WANT TO TAKE PART IN:

        BRITISH GRAND PRIX   2650MT   :1
        GERMAN  GRAND PRIX   1700MT   :2
        ITALIAN GRAND PRIX   2200MT   :3
        MONACO  GRAND PRIX   3100MT   :4

          ENTER A NUMBER [1 TO 4]
**********************************
          OK, THE GERMAN RACE
**********************************

OVER HOW MANY LAPS?
? 1
```

This time, I take fewer risks, and am soon on my way to completing the circuit without mishap:

```
ENGINE TEMPERATURE 117.5 C.  [MAX. 200]
BRAKE TEMPERATURE: 34.5 C.   [MAX. 500]
 DISTANCE COVERED: 1424.3 METERS
                 : .83 LAPS
YOU'RE IN POSITION 27
-------------------------------------
     CURRENT SPEED: 116.3 KPH
                  : 64.6 METERS PER MOVE
-------------------------------------
CORNER APPROACHING 75 METERS
RECOMMENDED SPEED: 200 KPH
-------------------------------------
```

A few moments later, and I've done it:

```
WELL DONE, HARTNELL!!
-------------------------------------
YOU MANAGED TO LAST OUT THE FULL
            1 LAP RACE...
-------------------------------------
```

The main value of this simulation, apart from that of its high entertainment potential, is showing how constraints and facts of 'real life' (such as the engine speed maximum, and the data on each

track) can be encoded into a program. You can also see the use of formulae to link the various outputs together (such as lines 37Ø and 38Ø which work out how far you have travelled, by converting your speed – in kilometres per hour – into metres) in a realistic way.

Note also that lines 32Ø and 33Ø, which act on your wishes to speed up or slow down, add or subtract a value which is related to the actual speed you are travelling. This means you can brake more dramatically when you are moving quickly than you can when moving slowly. Using a value which is related to current speed also overcomes the realism-destroying practice of just changing the current speed by a fixed value (such as 3.5, which I used in the first version of this program) each time a key is pressed.

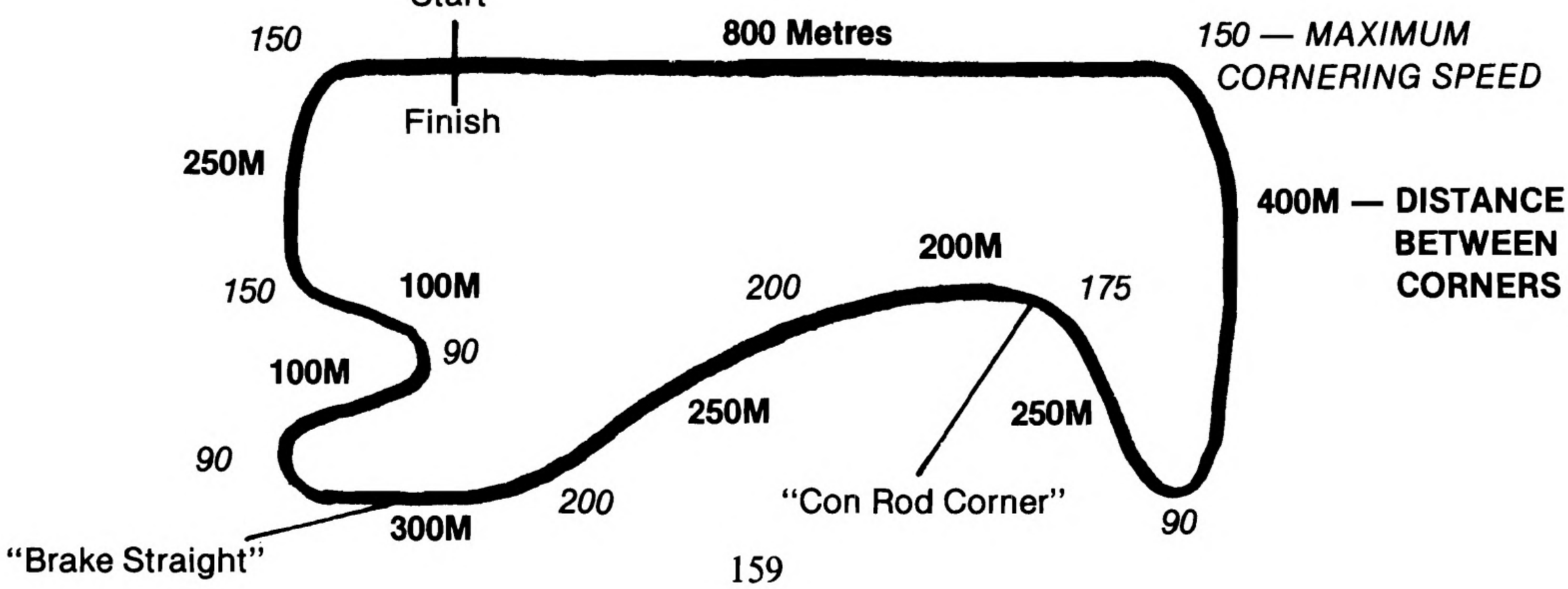
```
YOU FINISHED IN 30 POSITION,
AFTER STARTING IN 6TH POSITION...
YOUR AVERAGE SPEED WAS 100.38 KPH
YOU WERE 14 TH FASTEST ON STRAIGHTS,
AND 47 TH FASTEST ON CORNERS.

 PRESS 'S' FOR SAME RACE, 'N' FOR NEW
  RACE, 'E' TO END
```

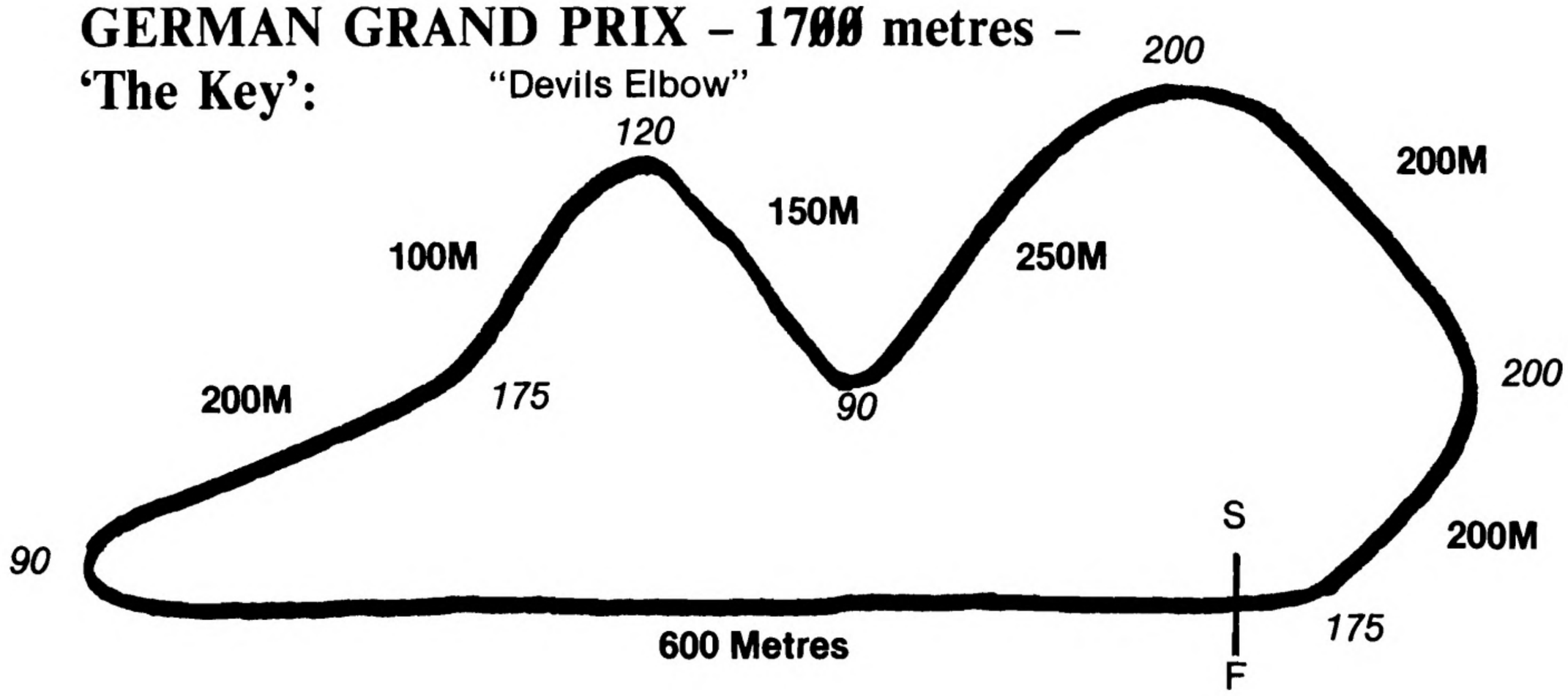
The Tracks

Here are the four Grand Prix circuits which are encoded within the program.

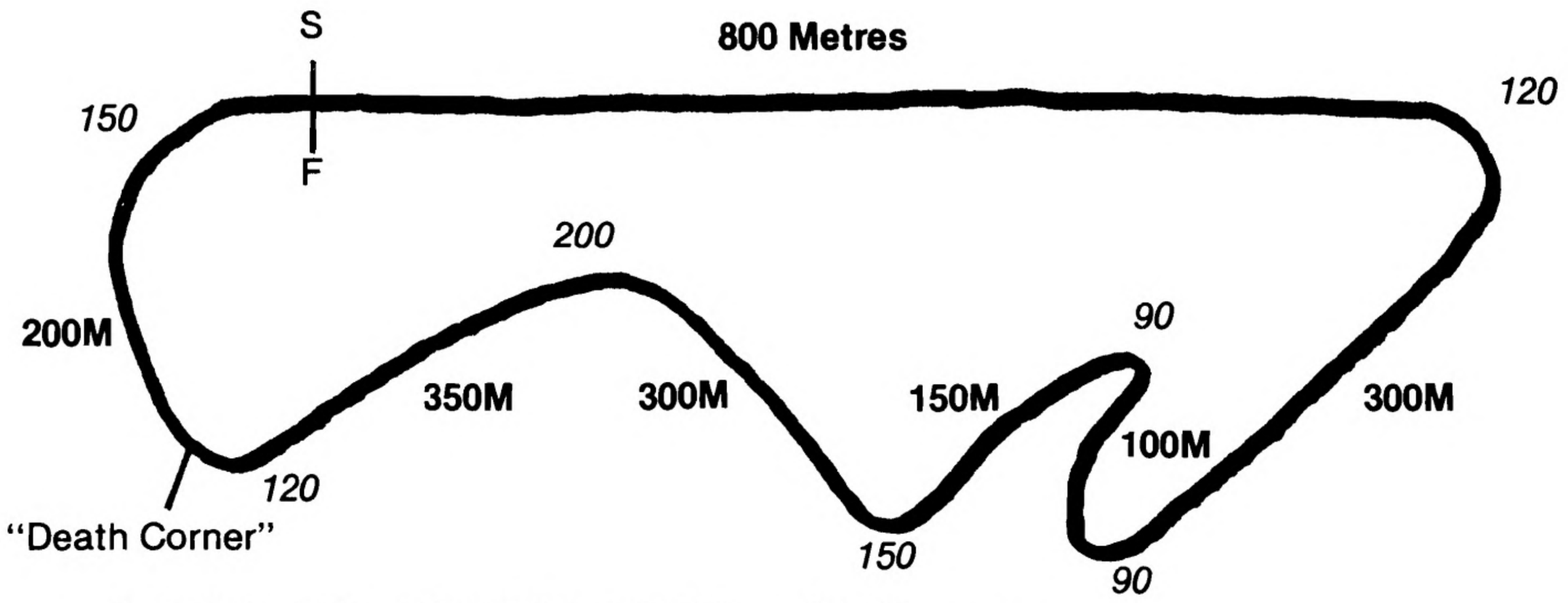
BRITISH GRAND PRIX – 265Ø metres – 'The Whale':



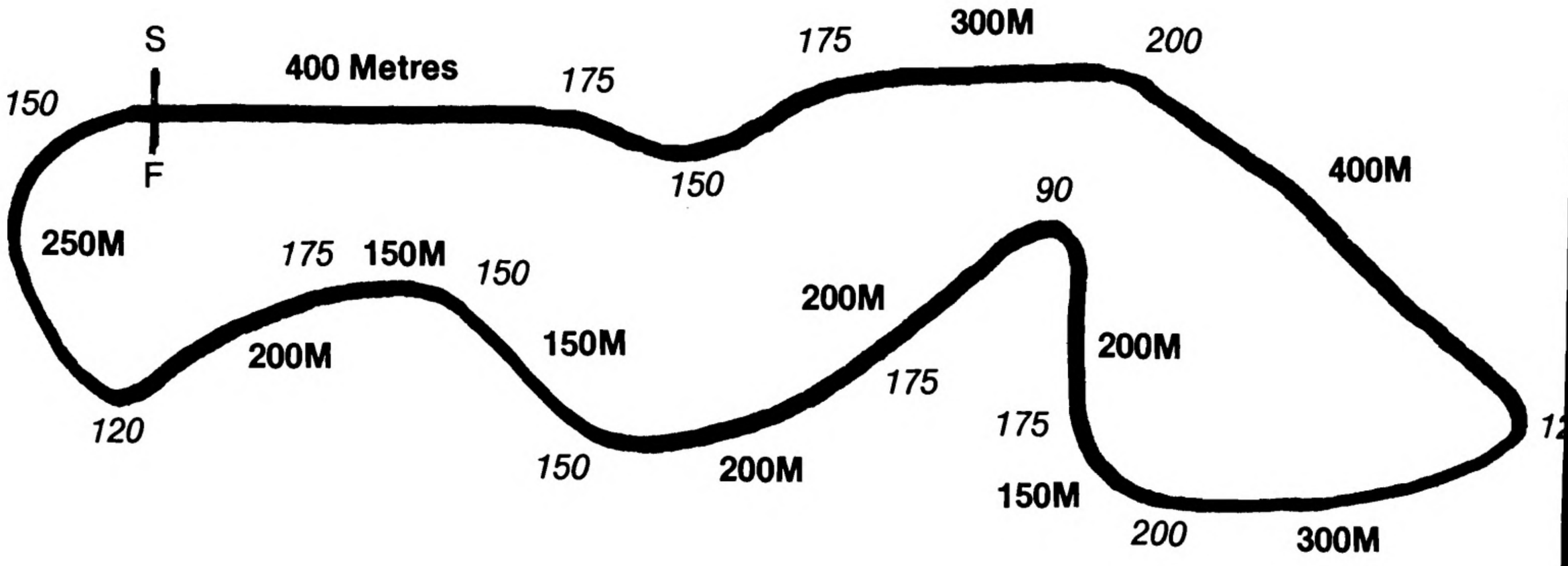
GERMAN GRAND PRIX – 17ØØ metres – 'The Key':



ITALIAN GRAND PRIX – 22ØØ metres – 'The Shoe':



MONACO GRAND PRIX – 31ØØ metres



Chapter Twenty-Two
The Grand Prix Listing

Take a firm grip on the wheel, and tackle the grand prix circuit with this listing:

```
10 REM GRAND PRIX
20 GOSUB 2200:REM INITIALIZE
30 GOSUB 1190:REM CHOOSE TRACK
40 REM ************************
50 REM MAJOR LOOP
60 GOSUB 120:REM PRINTOUT
70 GOSUB 280:REM ACCELERATION/CHECK
80 GOSUB 450:REM ENGINE/BRAKES
90 GOSUB 500:REM CORNER/POSITION
100 GOTO 60
110 REM ************************
120 REM PRINTOUT
130 CLS
140 PRINT "ENGINE TEMPERATURE"INT(ENG*10
)/10"C.  (MAX. 200)"
150 PRINT "BRAKE TEMPERATURE:"INT(BRAK*1
0)/10"C.  (MAX. 500)"
160 PRINT " DISTANCE COVERED:"INT(DIST*1
0)/10"METERS"
170 PRINT "                  :"INT(DIST*1
00/RR)/100"LAPS"
180 PRINT "YOU'RE IN POSITION"INT(FP)
190 PRINT "-----------------------------
-------"
200 PRINT "    CURRENT SPEED:"INT(SPEED*
10)/10"KPH"
210 PRINT "                  :"INT(SPEED*
5.555)/10"METERS PER MOVE"
220 PRINT "-----------------------------
-------"
```

```
230 PRINT "CORNER APPROACHING"INT[APP]"M
ETERS"
240 PRINT "RECOMMENDED SPEED:"C[C]"KPH"
250 PRINT "-----------------------------
--------"
260 RETURN
270 REM **************************
280 REM CHECK ACCELERATION AND FACTORS
290 X$=INKEY$:IF X$<>"Z" AND X$<>"M" AND
 X$<>" " THEN 290
300 PRINT TAB[12];"OK"
310 X=0
320 IF X$="M" THEN X=SPEED/15
330 IF X$="Z" THEN X=-SPEED/15
340 NUM=NUM+1:REM NUMBER OF MOVES
350 SPEED=SPEED+X
360 IF SPEED<0 THEN SPEED=0
370 TRAV=SPEED*.5555:REM DISTANCE
                         TRAVELLED
380 DIST=DIST+TRAV:REM TOTAL DISTANCE
                         TRAVELLED
390 ENG=ENG+[X/2]+.07:IF ENG<70 THEN ENG
=70+RND[1]*8:REM ENGINE TEMP
400 IF X>0 THEN BRAK=BRAK*.9:REM BRAKE
  TEMP FALLING; ACCELERATING
410 IF X<1 THEN BRAK=BRAK-[3*X]-RND[1]*3
:REM BRAKE TEMP INCREASING; BRAKING
420 IF BRAK<8 THEN BRAK=8+RND[1]*8
430 RETURN
440 REM **************************
450 REM CHECK ENGINE/BRAKE TEMP
460 IF ENG>200 THEN PRINT "YOUR ENGINE H
AS OVERHEATED":GOTO 830
470 IF BRAK>500 THEN PRINT "YOUR BRAKES
HAVE OVERHEATED":GOTO 830
480 RETURN
490 REM **************************
500 REM CHECK CORNERING SPEED
            AND FIELD POSITION
510 APP=APP-TRAV
520 IF APP>0 THEN RETURN
```

```
530 CRASH=0
540 IF SPEED>(C(C)*1.125) THEN CRASH=1:G
OTO 690
550 IF SPEED>(C(C)*1.1) THEN GOTO 690
560 PNT=PNT+100-((C(C)*1.1)-SPEED):REM C
ORNERING POINTS
570 NC=NC+1:REM NUMBER OF CORNERS
580 CP=96-(PNT/NC):REM CORNERING
                        POSITION
590 AM=AM+A(C):REM AVERAGE NUMBER
                    OF MOVES ALLOWED
600 RP=NUM-AM:REM RACING POSITION: YOUR
            MOVES MINUS AVERAGE MOVES
610 FP=(CP+RP)/2:REM FIELD POSITION IS
AVERAGE OF CORNER & RACE POSITIONS
620 IF FP<1 THEN FP=1
630 C=C+1
640 IF C=WW THEN C=1:REM LAP OVER
650 APP=APP+D(C)
660 IF LAP*QQ=AM THEN 910:REM RACE OVER
670 RETURN
680 REM ************************
690 REM CRASHED
700 CLS
710 PRINT "YOU CORNERED AT"INT(10*SPEED)
/10"KPH"
720 PRINT "AND THE MAXIMUM SPEED WAS JUS
T"C(C)
730 GOSUB 2330
740 PRINT "YOU SPIN OFF THE TRACK...
750 GOSUB 2330
760 IF CRASH=1 THEN 830
770 PRINT "YOU'VE LOST 20 SECONDS, BUT Y
OU'RE       ABLE TO REJOIN THE RACE"
780 NUM=NUM+10:SPEED=INT(2*C(C)/3)
790 PNT=PNT+50
800 GOSUB 2330
810 GOTO 570
820 REM ************************
830 PRINT ".........AND CRASH!!!!!"
840 PRINT "-----------------------------
-------"
```

```
850 PRINT "YOU ONLY COMPLETED"INT(10*DIS
T)/10"METERS,"
860 PRINT "OR"INT(DIST*100/RR)/100"LAPS
AND AT THAT"
870 PRINT "STAGE WERE IN POSITION"INT(FP
)
880 PRINT "-----------------------------
--------"
890 GOTO 1050
900 REM **************************
910 REM RACE OVER
920 CLS
930 EFLAG=1
940 FOR X=1 TO 20
950 PRINT TAB(X);"WELL DONE, ";A$;"!!"
960 PRINT TAB(21-X);"WELL DONE, ";A$;"!!
"
970 NEXT X
980 PRINT "-----------------------------
--------"
990 PRINT "YOU MANAGED TO LAST OUT THE F
ULL                    "LAP"LAP RACE..."
1000 PRINT "-----------------------------
--------"
1010 GOSUB 2330
1020 PRINT "YOU FINISHED IN POSITION"INT
(FP)
1030 PRINT "AFTER STARTING IN 6TH POSITI
ON..."
1040 GOSUB 2330
1050 PRINT "YOUR AVERAGE SPEED WAS"INT(D
IST*180/NUM)/100"KPH"
1060 GOSUB 2330
1070 IF RP<1 THEN RP=1
1080 IF CP<1 THEN CP=1
1090 PRINT "YOU WERE"INT(ABS(RP))"TH FAS
TEST ON STRAIGHTS,"
1100 PRINT "AND"INT(ABS(CP))"TH FASTEST
ON CORNERS."
1110 PRINT:PRINT "PRESS 'S' FOR SAME RAC
E, 'N' FOR NEW     RACE, 'E' TO END"
```

```
1120 I$=INKEY$:IF I$<>"S" AND I$<>"N" AN
D I$<>"E" THEN 1120
1130 IF I$="E" THEN END
1140 GOSUB 2240
1150 RESTORE
1160 IF I$="S" THEN GOSUB 1490:LAP=L2AP:
GOTO 60
1170 IF I$="N" THEN CLS:GOSUB 1250:GOTO
60
1180 REM **************************
1190 REM NAME AND TRACK DATA
1200 INPUT "WHAT IS YOUR NAME, DRIVER";A
$
1210 PRINT
1220 FOR X=1 TO 3
1230 PRINT TAB(4*X);"OK, GOOD LUCK, ";A$

1240 GOSUB 2330:NEXT X
1250 PRINT "**************************
**********"
1260 PRINT "WHICH RACE DO YOU WANT TO TA
KE PART IN:"
1270 PRINT
1280 PRINT TAB(7);"BRITISH GRAND PRIX  2
650MT  :1"
1290 PRINT TAB(7);"GERMAN  GRAND PRIX  1
700MT  :2"
1300 PRINT TAB(7);"ITALIAN GRAND PRIX  2
200MT  :3"
1310 PRINT TAB(7);"MONACO  GRAND PRIX  3
100MT  :4"
1320 PRINT
1330 PRINT TAB(7);"ENTER A NUMBER (1 TO
4)"
1340 K$=INKEY$
1350 IF K$<"1" OR K$>"4" THEN 1340
1360 GP=VAL(K$)
1370 PRINT "**************************
**********"
1380 PRINT TAB(8);"OK, THE ";
1390 IF GP=1 THEN PRINT "BRITISH";
```

```
1400 IF GP=2 THEN PRINT "GERMAN";
1410 IF GP=3 THEN PRINT "ITALIAN";
1420 IF GP=4 THEN PRINT "MONACO";
1430 PRINT " RACE"
1440 PRINT "**********************************
**********"
1450 PRINT:PRINT "OVER HOW MANY LAPS?"
1460 INPUT LAP:IF LAP<1 THEN 1460
1470 LAP=INT(LAP+.5):L2AP=LAP
1480 REM **************************
1490 REM BRITISH DATA
1500 SPEED=140
1510 FOR X=1 TO 9
1520 READ D(X):REM DISTANCE BETWEEN
                              CORNERS
1530 NEXT X
1540 DATA 800,400,250,200,250,300,100,10
0,250
1550 FOR X=1 TO 9
1560 READ C(X):REM RECOMMENDED
                   MAXIMUM CORNERING SPEED
1570 NEXT X
1580 DATA 150,90,175,200,200,90,90,150,1
50
1590 FOR X=1 TO 9
1600 READ A(X):REM AVERAGE NUMBER OF
        MOVES ALLOWED BETWEEN CORNERS
1610 NEXT X
1620 DATA 8,4,2,2,2,2,1,1,2
1630 APP=800:WW=10:QQ=24:RR=2650
1640 IF GP=1 THEN RETURN
1650 REM **************************
1660 REM GERMAN DATA
1670 SPEED=85
1680 FOR X=1 TO 7
1690 READ D(X)
1700 NEXT X
1710 DATA 600,200,100,150,250,200,200
1720 FOR X=1 TO 7
1730 READ C(X)
1740 NEXT X
```

```
1750 DATA 90,175,120,90,200,200,175
1760 FOR X=1 TO 7
1770 READ A[X]
1780 NEXT X
1790 DATA 6,2,1,2,2,2,2
1800 APP=600:WW=8:QQ=17:RR=1700
1810 IF GP=2 THEN RETURN
1820 REM **************************
1830 REM ITALIAN DATA
1840 SPEED=108
1850 FOR X=1 TO 7
1860 READ D[X]
1870 NEXT X
1880 DATA 800,300,100,150,300,350,200
1890 FOR X=1 TO 7
1900 READ C[X]
1910 NEXT X
1920 DATA 120,90,90,150,200,120,150
1930 FOR X=1 TO 7
1940 READ A[X]
1950 NEXT X
1960 DATA 8,3,2,1,3,3,2
1970 APP=800:WW=8:QQ=22:RR=2200
1980 IF GP=3 THEN RETURN
1990 REM **************************
2000 REM MONACO DATA
2010 SPEED=162.5
2020 FOR X=1 TO 14
2030 READ D[X]
2040 NEXT X
2050 DATA 400,100,100,300,400,300,150,20
0,200,200
2060 DATA 150,150,200,250
2070 FOR X=1 TO 14
2080 READ C[X]
2090 NEXT X
2100 DATA 175,150,175,200,120,200,175,90
,175,150
2110 DATA 150,175,120,150
2120 FOR X=1 TO 14
2130 READ A[X]
```

```
2140 NEXT X
2150 DATA 4,1,1,3,4,3,1,2,2,2
2160 DATA 1,2,2,2
2170 APP=400:WW=15:QQ=30:RR=3100
2180 RETURN
2190 REM ************************
2200 REM INITIALIZATION
2210 CLS
2220 RANDOMIZE VAL[RIGHT$[TIME$,2]]
2230 DIM A[14],C[14],D[14]
2240 C=1:FP=6:PNT=0:NC=0:CP=0
2250 AM=0:RP=0:APP=0
2260 NUM=0:REM NUMBER OF MOVES
2270 ENG=100:BRAK=10:TRAV=0:DIST=0
2280 EFLAG=0
2290 X=0
2300 RETURN
2310 REM ************************
2320 REM DELAY
2330 FOR O=1 TO 1000:NEXT O
2340 RETURN
```

Chapter Twenty-Three
Up, Up and Away

Flight simulators are probably the most entertaining simulation programs you can run. Many of us will never learn to fly, but would like to enjoy the experience. A flight simulation program gives you a chance to find out a little of how it feels to fly.

The main challenge facing a pilot is to keep on top of a number of situations which are changing all at once. Ignore any one of them – such as the angle at which the plane is flying, or the amount of fuel you have – for too long, and your plane may well drop out of the sky. Although this program does not occur in real time, so you have a chance to think about the choices facing you before you act, you'll still find it a fairly challenging (and frustrating) simulation to master.

Even getting the plane off the ground is not particularly easy the first few times you run the program. Get it in the air, and you have to fight every second to keep it there.

And once you try and land . . .

Crashing during a landing attempt is a trifle inconvenient in a real plane. The moment you try and land 15Ø feet above the ground, you'll give thanks that you are flying your computer.

Mastering the Controls

This simulation is the most complex in the book. It will take you a little longer than the others to learn to use. This longer learning time will, however, be amply repaid as you're sure to find this program the most satisfying to run. *Flight Simulation* contains

another bonus which makes it very worthwhile to run. Your flight is automatically recorded, so at any moment while you're still in the air, you can tell the computer you want to see the whole of the current flight from the beginning again. It is very entertaining to watch your flight unfolding quickly in front of you. The controls are handed back to you once you reach the point in the original flight where you told the computer you wanted a repeat.

Think of the flight as having three parts, each of which requires a different approach to the controls. Part one is the takeoff, part two is the actual flight, and part three is the landing. You'll become an ace pilot fairly easily if you concentrate on mastering each flight section in turn.

And don't give up, even after your twentieth attempt at landing has ended in disaster. The satisfaction you'll feel when you manage a perfect landing for the first time will more than compensate for all the frustration inherent in the learning process.

This is what you see when you first run the program:

```
          HORIZON                    HEADING
:--------------------:----------:
:                    :    .N.   :
:                    :   ..@..  :
:                    : .. : .. :
:  ****************  : W--X--E :
:                    : .. : .. :
:                    :   ..:..  :
:                    :    .S.   :
:-----------------------------:
:RANGE 0  : TIME .1  : 226
:-----------------------------:
:AIRSPEED  : 0
:>
:ALTIMETER: 0          0 DEG.
:>
:FUEL      : 750
:-------------------->
:-----------------------------:
:ELEVATION: 0  : -------
:    > UNDERCARRIAGE DOWN <  :
```

In the top left is the horizon. This can tilt rather alarmingly from side to side as you turn the plane, or go to the bottom of the 'windshield' as you climb, or to the top when you descend. It may take a short while in order for you to be able to 'see' this line as the horizon, but once you do, you'll find it conveys a much better impression of 'being in the air' than you might imagine at this stage.

To the left of the horizon is your compass. The '@' shows the direction you're heading. You always start a flight facing due north, and must take off more or less in this direction. Your exact heading is shown further down the instrument cluster, in the same line as ALTIMETER, where you'll see 'Ø DEG.'.

The line of readings underneath the horizon and compass shows the distance over the ground you've covered so far (the 'range'), the time of the flight, and the direction in which you must be flying in order to land (assuming you get this baby up in the air, and keep it there for a few seconds!). You need to be within 12 degrees of this heading for a successful landing. The required heading changes from run to run, but does not change during a flight.

Under this is your airspeed, altimeter (height above the ground) and remaining fuel. The line of dashes ending in a greater than sign (____>) indicates the magnitude of each of these readings. That is, the line gets longer as you go faster, or climb higher.

The elevation is the angle of your plane against the ground. When you enter a command, by touching a single key, a full-word version of the command appears in the position currently occupied by the series of dashes you can see after the elevation angle in the sample run above. Finally, you are told whether or not your undercarriage is down.

After a few inputs, the screen looks like this:

```
   HORIZON                HEADING
:-------------------:---------:
:                   :    .N.  :
:                   :  ..●..  :
:                   : ..  :  .. :
:  ***************  : W--X--E :
:                   : ..  :  .. :
:                   :  ..:..  :
:                   :    .S.  :
:-----------------------------:
:RANGE .6  : TIME 1.3  : 226
:-----------------------------:
:AIRSPEED  : 30
:->
:ALTIMETER: 0        0 DEG.
:>
:FUEL      : 692
:-------------------->
:-----------------------------:
:ELEVATION: 20  : NOSE UP
:    > UNDERCARRIAGE DOWN <  :
```

Your fuel has gone down slightly, your airspeed is up to 3Ø, and the nose of the plane is pointing 2Ø degrees into the air. You can see, from down near the bottom right hand corner of the screen, that NOSE UP is the full equivalent of the most recent command you entered.

A little later, and we are in the air:

```
   HORIZON                HEADING
:-------------------:---------:
:                   :    .N.  :
:                   :  ..●..  :
:                   : ..  :  .. :
:                   : W--X--E :
:  ***************  : ..  :  .. :
:                   :  ..:..  :
:                   :    .S.  :
:-----------------------------:
:RANGE 1   : TIME 2   : 226
:-----------------------------:
:AIRSPEED  : 51
:-->
:ALTIMETER: 66       359 DEG.
:-->
:FUEL      : 660
:-------------------->
:-----------------------------:
:ELEVATION: 19  : THROTTLE ON
:    > UNDERCARRIAGE DOWN <  :
```

You need an airspeed of between 45 and 6Ø, and an elevation greater than 1Ø, in order to get off the ground. Let's try turning:

```
   HORIZON                HEADING
:-------------------:---------:
:                   :    .N.  :
:                   :  .@:..  :
:                   : .. : .. :
: ****              : W--X--E :
:     ******        : .. : .. :
:           *****   :  ..:..  :
:                   :   .S.   :
:-----------------------------:
:RANGE 1.2 : TIME 2.3 : 226
:-----------------------------:
:AIRSPEED  : 40
:-->
:ALTIMETER: 143     344 DEG.
:----->
:FUEL      : 648
:--------------------->
:-----------------------------:
:ELEVATION: 15 : BANK LEFT
:     > UNDERCARRIAGE UP <    :
```

Note that the undercarriage is now up (which saves on fuel, and decreases drag on the plane) and the 'horizon' is now sloping to down to the right, as the plane banks left. It is now heading at 344 degrees, and the @ symbol on the compass has moved to show the approximate direction in which the plane is flying.

```
   HORIZON                HEADING
:-------------------:---------:
:                   :    .N.  :
:                **   :  ..@..  :
:             ***     : .. : .. :
:          ***        : W--X--E :
:       ***           : .. : .. :
:  ***                :  ..:..  :
:                     :   .S.   :
:-----------------------------:
:RANGE 3.1 : TIME 3.7 : 226
:-----------------------------:
:AIRSPEED  : 59
:-->
:ALTIMETER: 353     3 DEG.
:----------->
:FUEL      : 590
:--------------------->
:-----------------------------:
:ELEVATION: 7 : BANK RIGHT
:     > UNDERCARRIAGE UP <    :
```

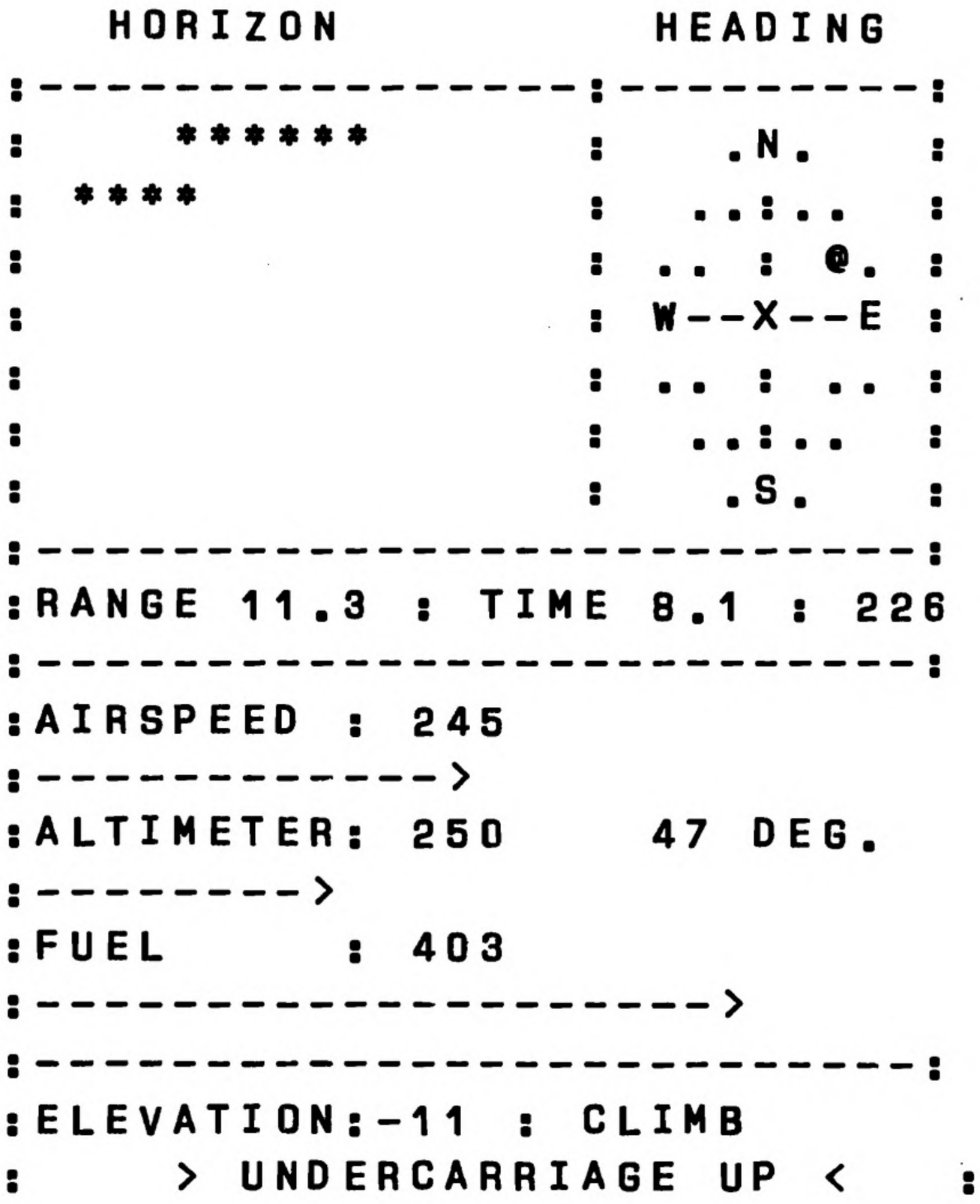
In this 'snapshot', taken a few moments later, we have started to bank to the right. Our airspeed is up to 59, and we've continued to climb. However, the elevation has fallen, so the rate of climb will decrease. As we level off, the horizon gradually tilts back to the horizontal:

```
   HORIZON                HEADING
:-------------------:---------:
:                   :    .N.  :
:                   :   ..:@.  :
:                   :  ..  :  .. :
:           *****   : W--X--E  :
:     ******        :  ..  :  .. :
:  ****             :    ..:..  :
:                   :     .S.   :
:---------------------------------:
:RANGE 3.6  : TIME 5.3  : 226
:---------------------------------:
:AIRSPEED   : 76
:--->
:ALTIMETER: 475      15 DEG.
:---------------->
:FUEL       : 525
:---------------------->
:---------------------------------:
:ELEVATION: 12  : THROTTLE ON
:     > UNDERCARRIAGE UP <      :
```

As I said in the introduction, there are an awful lot of things to keep under control. While concentrating on the direction in which I'm flying, I've been neglecting the plane's elevation which has been slowly falling:

```
   HORIZON                HEADING
:-------------------:---------:
:                   :    .N.  :
:                   :   ..:@.  :
:                   :  ..  :  .. :
:           *****   : W--X--E  :
:     ******        :  ..  :  .. :
:  ****             :    ..:..  :
:                   :     .S.   :
:---------------------------------:
:RANGE 7.4  : TIME 6.9  : 226
:---------------------------------:
:AIRSPEED   : 152
:-------->
:ALTIMETER: 593      31 DEG.
:---------------------->
:FUEL       : 457
:------------------------>
:---------------------------------:
:ELEVATION: 0   : THROTTLE ON
:     > UNDERCARRIAGE UP <      :
```

Suddenly I notice the nose is pointing downward (note how the horizon is now right at the top) and frantically try to climb:



However, too much is happening too quickly, and there is an inevitable result:

CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!
CRASH!

The final picture is pretty bleak:

```
          HORIZON                    HEADING
        :--------------------:----------:
        :  C  R**   A  **  S* :     .N.    :
        :  *   A  **  S*  H*  :    ..:..   :
        :  R**   A  **  S*  H :  ..  :  ..  :
        :  R**   A  **  S*  H :  W--X-@E   :
        :    A  **  S*  H*  ! :  ..  :  ..  :
        :  **   A  **  S*  H* :    ..:..   :
        :  *  *C  R**   A  ** :     .S.    :
        :--------------------------------:
        :RANGE 14.1  : TIME 9.2  : 226
        :--------------------------------:
        :AIRSPEED  : 273
        :-------------->
        :ALTIMETER:  0          84 DEG.
        :>
        :FUEL      : 366
        :---------------------->
        :--------------------------------:
        :ELEVATION:-16  : THROTTLE ON
        :      > UNDERCARRIAGE UP <      :
```

The commands you have at your disposal are:

R – this allows you to repeat the current flight from the beginning up to that point
Space bar – this is used for 'throttle on', and increases your speed
. – use this to throttle back
Q – to increase elevation
A – to decrease elevation
Z – to turn left
M – to turn right
1 – to change undercarriage from up to down, or from down to up.

Remember, to take off you must have a speed between 45 and 6Ø, and your elevation must be greater than 1Ø. To land, your undercarriage must be down (!), your heading must be within 12

degrees of the setting shown beneath the compass, you must be 15 or below on your altimeter, and your speed must not be greater than 2Ø. You'll find landing is even harder (as in real life) than is taking off.

The complete *Flight Simulation* listing is in the next chapter.

Chapter Twenty-Four
The Flight Simulation Listing

Time now to earn your wings, with this listing:

```
10 REM FLIGHT SIMULATION
20 RPT=0
30 LD=INT(RND(1)*360)
40 DIM E$(1000):REM THIS HOLDS FLIGHT
                              RECORD
50 DIM A$(7),C$(7):REM THESE ARRAYS
 HOLD HORIZON AND COMPASS OUTPUT
60 REM ************************
70 GOSUB 2320:REM INITIALIZE
80 IF CRASH=0 THEN GOSUB 820:REM HORIZON
                                /COMPASS
90 GOSUB 500:REM PRINTOUT
100 IF CRASH=1 THEN END
110 IF LAND=1 AND UFLAG=1 THEN PRINT "WE
LL DONE. A PERFECT LANDING!!":END
120 IF LAND=1 AND UFLAG=0 THEN PRINT "YO
UR WHEELS ARE UP":GOSUB 1780:GOTO 90
130 T=AIRSPEED:STALL=0
140 X$=INKEY$
150 IF X$="R" THEN RPT=1:GOTO 70
160 IF RPT=1 AND E$(CLOCK+1)="" THEN RPT
=0:GOTO 140
170 IF RPT=1 THEN X$=E$(CLOCK+1)
180 IF X$="" THEN 140
190 IF CLOCK<999 THEN E$(CLOCK+1)=X$
200 IF TAKEOV=1 THEN ELEVATE=INT(ELEVATE
+RND(1)*2-RND(1)*3)
210 IF AIRSPEED<3 THEN 290
220 IF X$="Q" THEN ELEVATE=ELEVATE+5:EFL
AG=5:IF ELEVATE>60 THEN STALL=1
```

```
230 IF X$="A" THEN ELEVATE=ELEVATE-5:EFL
AG=-5:IF ELEVATE<-70 THEN STALL=-1
240 IF STALL<>0 THEN GOSUB 1640
250 IF ALTIMETER<1 THEN 290:REM PREVENTS
      DRAMATIC TURNS ON THE GROUND
260 IF X$="Z" THEN WA=WA-.5:ANG=ANG-6:IF
 WA<-3 THEN WA=-3
270 IF X$="M" THEN WA=WA+.5:ANG=ANG+6:IF
 WA>3 THEN WA=3
280 ANG=INT(ANG+RND(1)*2-RND(1)*2)
290 IF X$=" " THEN AIRSPEED=AIRSPEED+8.5

300 IF X$="." THEN AIRSPEED=AIRSPEED-7
310 AIRSPEED=AIRSPEED-ELEVATE/5
320 IF UFLAG=1 THEN AIRSPEED=AIRSPEED-1.
5:FUEL=FUEL-.5
330 IF AIRSPEED<0 THEN AIRSPEED=0
340 IF AIRSPEED>400 THEN AIRSPEED=400
350 IF X$="1" AND UFLAG=0 THEN UFLAG=1:G
OTO 370
360 IF X$="1" AND UFLAG=1 THEN UFLAG=0
370 FUEL=FUEL-(ABS(T-AIRSPEED)/10)-3.75
380 IF FUEL<1 THEN GOSUB 1780
390 IF TAKEOV=1 THEN 420
400 IF ELEVATE>10 AND AIRSPEED>45 AND AI
RSPEED<60 AND UFLAG=1 THEN TAKEOV=1
410 IF TAKEOV=0 THEN ALTIMETER=0:GOTO 45
0
420 IF LAND=0 AND AIRSPEED<30 THEN ELEVA
TE=ELEVATE-5:ALTIMETER=9*ALTIMETER/10
430 ALTIMETER=ALTIMETER+INT((((ELEVATE+.1
)*AIRSPEED)+EFLAG*AIRSPEED/1000)/80
440 IF ALTIMETER<300 AND TAKEOV=1 THEN A
LTIMETER=ALTIMETER+AIRSPEED/30+ELEVATE
450 IF ALTIMETER<0 THEN GOSUB 1780:REM
      CRASH
460 REM CHANGE NEXT TWO LINES TO MAKE IT
        EASIER (OR EVEN HARDER) TO LAND
470 IF ALTIMETER>15 AND AIRSPEED>20 OR T
AKEOV=0 THEN 80
480 IF ABS(ANG-LD)<13 OR ABS(ANG+360-LD)
<13 THEN LAND=1 :GOTO 80
```

```
490 REM **************************
500 REM PRINTOUT
510 CLS
520 PRINT "    HORIZON";TAB(20);"HEADING"
530 EV=INT(ELEVATE/10)
540 IF EV>2 THEN EV=2
550 IF EV<-2 THEN EV=-2
560 IF EV<>0 AND TAKEOV=1 AND CRASH=0 TH
EN GOSUB 1920
570 PRINT ":------------------:---------:"
580 FOR J=1 TO 7
590 PRINT ": ";A$(J);" :";C$(J);" :"
600 A$(J)=""
610 NEXT J
620 PRINT ":-------------------------:"
630 DIST=DIST+ABS((COS(ELEVATE))*AIRSPEE
D)/360
640 CLOCK=CLOCK+1
650 PRINT ":RANGE"INT(DIST*10)/10": TIME
"INT(CLOCK)/10;":"LD
660 PRINT ":-------------------------:"
670 PRINT ":AIRSPEED :"INT(AIRSPEED)
680 PRINT ":";LEFT$(Q$,INT(AIRSPEED/20))
;">"
690 PRINT ":ALTIMETER:"INT(ALTIMETER);
700 IF ANG<0 THEN PRINT TAB(19);360+ANG"
DEG."
710 IF ANG>=0 THEN PRINT TAB(19);ANG"DEG
."
720 MR=INT(ALTIMETER/30):IF MR>20 THEN M
R=20
730 PRINT ":";LEFT$(Q$,MR);">"
740 PRINT ":FUEL      :"INT(FUEL)
750 PRINT ":";LEFT$(Q$,20-INT(FUEL/750))
;">"
760 PRINT ":-------------------------:"
770 PRINT ":ELEVATION:"ELEVATE":  ";:GOSU
B 2210:PRINT U$
780 IF UFLAG=1 THEN PRINT ":";TAB(5);">
UNDERCARRIAGE DOWN <  :"
```

```
790 IF UFLAG=0 THEN PRINT ":";TAB(6);">
UNDERCARRIAGE UP <    :"
800 RETURN
810 REM ***************************
820 REM ASSIGN HORIZON/COMPASS
830 IF ABS(INT(WA+.5))=3 THEN GOSUB 980
840 IF ABS(INT(WA+.5))=2 THEN GOSUB 1070

850 IF ABS(INT(WA+.5))=1 THEN GOSUB 1160

860 IF INT(WA+.5)=0 THEN GOSUB 1250
870 REM NEXT TWO LINES USED TO
    GRADUALLY STRAIGHTEN UP WINGS
880 IF WA>0 THEN WA=WA-.2
890 IF WA<0 THEN WA=WA+.2
900 IF WA>.2 THEN 1350
910 FOR Z=1 TO 7
920 M$(8-Z)=A$(Z)
930 NEXT Z
940 FOR Z=1 TO 7
950 A$(Z)=M$(Z)
960 NEXT Z
970 GOTO 1350
980 REM WA=3 OR -3
990 A$(1)="                  *  "
1000  A$(2)="                **   "
1010  A$(3)="              **     "
1020  A$(4)="           **        "
1030  A$(5)="        **           "
1040  A$(6)="    **               "
1050  A$(7)="**                  "
1060  RETURN
1070  REM WA=2 OR -2
1080  A$(1)="                     "
1090  A$(2)="                  **  "
1100  A$(3)="               ***    "
1110  A$(4)="          ***         "
1120  A$(5)="     ***              "
1130  A$(6)="***                   "
1140  A$(7)="                      "
1150  RETURN
```

```
1160 REM WA=1 OR -1
1170 A$[1]="                    "
1180 A$[2]="                    "
1190 A$[3]="              *****"
1200 A$[4]="    ******          "
1210 A$[5]="****                "
1220 A$[6]="                    "
1230 A$[7]="                    "
1240 RETURN
1250 REM WA=0
1260 A$[1]="                    "
1270 A$[2]="                    "
1280 A$[3]="                    "
1290 A$[4]="******************"
1300 A$[5]="                    "
1310 A$[6]="                    "
1320 A$[7]="                    "
1330 RETURN
1340 REM ************************
1350 REM ASSIGN COMPASS STRINGS
1360 F2=ANG-F1
1370 IF F2<0 THEN FA=INT[[F2+375]/30]
1380 IF F2>=0 THEN FA=INT[[F2+15]/30]
1390 IF FA=12 THEN FA=0
1400 C$[1]="    .N.   "
1410 IF FA=11 THEN C$[2]="   .@:..  ":GOTO
 1450
1420 IF FA=0 THEN C$[2]="   ..@..  ":GOTO
1450
1430 IF FA=1 THEN C$[2]="   ..:@.  ":GOTO
1450
1440 C$[2]="   ..:..  "
1450 IF FA=10 THEN C$[3]= "  .@ :  .. ":GOT
0 1480
1460 IF FA=2 THEN C$[3]= "  .. :  @. ":GOTO
 1480
1470 C$[3]="  .. :  .. "
1480 IF FA=9 THEN C$[4]=" W@-X--E":GOTO
1510
1490 IF FA=3 THEN C$[4]=" W--X-@E":GOTO
1510
```

```
1500 C$[4]="  W--X--E"
1510 IF FA=8 THEN C$[5]="  .@  :  .."":GOTO
     1540
1520 IF FA=4 THEN C$[5]="  ..  :  @."":GOTO
     1540
1530 C$[5]="  ..  :  .."
1540 IF FA=7 THEN C$[6]="    .@:..  "":GOTO
     1580
1550 IF FA=6 THEN C$[6]="    ..@..  "":GOTO
     1580
1560 IF FA=5 THEN C$[6]="    ..:@.  "":GOTO
     1580
1570 C$[6]="    ..:..  "
1580 C$[7]="     .S.   "
1590 IF ANG>360 THEN ANG=ANG-360
1600 F2=ANG
1610 IF W>0 THEN W=W-.4
1620 IF W<0 THEN W=W+.4
1630 RETURN
1640 REM STALL/FALL
1650 IF STALL=-1 THEN 1710
1660 FOR J=1 TO 10
1670 PRINT TAB[J];"YOU HAVE STALLED!"
1680 NEXT J
1690 AIRSPEED=AIRSPEED/4
1700 RETURN
1710 FOR J=1 TO 10
1720 PRINT TAB[J];"UNCONTROLLED DIVE!!"
1730 PRINT TAB[21-J];"PULL UP!!"
1740 NEXT J
1750 ALTIMETER=4*ALTIMETER/5
1760 RETURN
1770 REM **************************
1780 REM CRASH
1790 CRASH=1
1800 ALTIMETER=0
1810 M$="** *C R**   A ** S* H* !!*":REM
     25 CHARACTERS LONG
1820 FOR J=1 TO 20
1830 PRINT TAB[J];"CRASH!"
1840 PRINT TAB[21-J];"CRASH!"
```

```
1850 NEXT J
1860 FOR J=1 TO 7
1870 G=INT(RND(1)*11)+1
1880 A$(J)=MID$(M$,G,14)
1890 NEXT J
1900 RETURN
1910 REM ***************************
1920 REM ADJUST HORIZON
1930 G$="              ":REM 14 SPACES
1940 ON EV+3 GOSUB 1960,2020,2070,2080,2
140
1950 RETURN
1960 REM EV=-2
1970 FOR J=1 TO 4
1980 A$(J)=A$(J+3)
1990 NEXT J
2000 A$(5)=G$:A$(6)=G$:A$(7)=G$
2010 RETURN
2020 REM EV=-1
2030 FOR J=1 TO 6
2040 A$(J)=A$(J+1)
2050 NEXT J
2060 A$(7)=G$
2070 RETURN:REM EV=0
2080 REM EV=1
2090 FOR J=6 TO 1 STEP -1
2100 A$(J+1)=A$(J)
2110 NEXT J
2120 A$(1)=G$
2130 RETURN
2140 REM EV=2
2150 FOR J=4 TO 1 STEP -1
2160 A$(J+3)=A$(J)
2170 NEXT J
2180 A$(1)=G$:A$(2)=G$:A$(3)=G$
2190 RETURN
2200 REM ***************************
2210 REM INPUT INTO COMMAND NAME
2220 U$="--------"
2230 IF X$=" " THEN U$="THROTTLE ON"
2240 IF X$="." THEN U$="THROTTLE OFF"
```

```
2250 IF X$="Q" AND ALTIMETER>0 THEN U$="
CLIMB"
2260 IF X$="Q" AND ALTIMETER=0 THEN U$="
NOSE UP"
2270 IF X$="A" THEN U$="NOSE DOWN"
2280 IF X$="Z" THEN U$="BANK LEFT"
2290 IF X$="M" THEN U$="BANK RIGHT"
2300 RETURN
2310 REM ************************
2320 REM INITIALIZATION
2330 CLS
2340 RANDOMIZE VAL(RIGHT$(TIME$,2))
2350 Q$="---------------------":REM
    21 CHARACTERS IN STRING
2360 UFLAG=1:REM UNDERCARRIAGE -
                  1 - DOWN, 0 - UP
2370 EFLAG=0:REM CLIMB RATE
2380 ANG=0:TAKEOV=0:LAND=0
2390 AIRSPEED=0
2400 DIST=0:REM DISTANCE COVERED 'RANGE'

2410 ALTIMETER=0
2420 ELEVATE=0:REM ANGLE OF ELEVATION
2430 WA=0:REM 'WING ANGLE; USED IN
                        HORIZON PRINTOUT
2440 FUEL=750:CRASH=0:F2=0:F1=0:REM
 FOR DIRECTION CHANGE/COMPASS ROUTINE
2450 CLOCK=0:REM TIME
2460 X$=""
2470 RETURN
```

Commodore 64 Listings

Space Landing

```
10 REM SPACE LANDING SIMULATION
20 FLAG=INT(RND(VAL(RIGHT$(TI$,2)))):REM
RANDOMIZE
30 REM ********************************
40 REM SET STARTING VALUES
45 POKE 53280,0:POKE 53281,0
50 FUEL=200+RND(1)*40
60 VELOCITY=RND(1)*20-6
70 HEIGHT=500-RND(1)*10
80 PRINT"
{CLR}{WHT}"
90 PRINT" FUEL";TAB(12);" VELOCITY";TAB(
24);" HEIGHT"
100 REM ********************************
110 REM MAJOR CYCLE
120 GOSUB 430
130 IF FUEL<=0 THEN FUEL=0:THRUST=0:GOTO
 170
140 GET A$
150 IF A$<"0" OR A$>"9" THEN 140
160 THRUST=VAL(A$)+.1
170 FUEL=FUEL-THRUST
180 FLAG=THRUST-2
190 THRUST=0
200 HEIGHT=HEIGHT+VELOCITY+FLAG/4
210 VELOCITY=VELOCITY+FLAG
220 IF HEIGHT<=10 THEN 240
230 IF HEIGHT>10 THEN 120
240 IF VELOCITY>-9 AND VELOCITY<5 THEN 2
90
250 GOSUB 410
260 PRINT"YOU HAVE CRASHED INTO THE SURF
ACE..."
270 IF HEIGHT>0 THEN HEIGHT=-HEIGHT
280 GOTO 320
```

```
290 GOSUB 4000:PRINT"YOU HAVE LANDED SAF
ELY!"
300 PRINT"YOUR SKILL RATING IS"INT(-1000
*FUEL/(VELOCITY-HEIGHT))
310 HEIGHT=0
320 GOSUB 410
330 PRINT"FINAL INSTRUMENT READINGS WERE
:"
340 PRINT" FUEL";TAB(12);" VELOCITY";TAB
(24);" HEIGHT"
350 GOSUB 430
360 GOSUB 410
370 IF HEIGHT>=0 THEN END
380 PRINT"NEW CRATER ON MOON"INT(ABS(100
*(HEIGHT+.2)/3))/100"METERS DEEP!"
390 PRINT"YOUR SKILL RATING IS "INT(100*
FUEL/(VELOCITY-HEIGHT))
400 END
410 PRINT"-------------------------------
---------"
420 RETURN
430 PRINT INT(100*FUEL)/100;
440 PRINT TAB(12);-INT(100*VELOCITY)/100
;
450 IF HEIGHT>=0 THEN PRINT TAB(24);INT(
100*HEIGHT)/100
460 IF HEIGHT<0 THEN PRINT
470 RETURN
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP .7
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200
4120 POKE SID+1,L1
```

```
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -1
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

Monte Carlo

```
10 REM MONTE CARLO DEMONSTRATION
20 GOSUB 370:REM INITIALISE
30 REM ************************
40 REM MAJOR CYCLE
45 PRINT"
{CLR}"
50 GOSUB 100:REM PRINT
60 IF P=EP AND Q=EQ THEN PRINT:PRINT"DEM
ONSTRATION OVER":END
70 GOSUB 230:REM GENERATE MOVE
80 GOTO 50
90 REM ************************
100 REM PRINTOUT
110 A$(P,Q)="O"
120 M=M+1
130 PRINT"
{HOME}":PRINT:PRINT
140 PRINT"MOVE"M
150 FOR X=1 TO 10
160 FOR Y=1 TO 10
170 PRINTA$(X,Y);" ";
180 NEXT Y
190 PRINT
200 NEXT X
210 RETURN
220 REM ************************
230 REM GENERATE MOVE
240 A$(P,Q)="."
250 G=0
260 T=INT(RND(1)*4)+1
270 ON T GOSUB 310,320,330,340
280 IF G=0 THEN 260
290 IF G=1 AND RND(1)>.5 THEN 260
300 RETURN
310 IF P>1 THEN P=P-1:G=G+1:RETURN
320 IF P<10 THEN P=P+1:G=G+1:RETURN
330 IF Q>1 THEN Q=Q-1:G=G+1:RETURN
340 IF Q<10 THEN Q=Q+1:G=G+1:RETURN
```

```
350 RETURN
360 REM **************************
370 REM INITIALISE
380 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}"
390 X=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
400 DIM A$(10,10)
410 FOR X=1 TO 10
420 FOR Y=1 TO 10
430 A$(X,Y)="."
440 NEXT Y
450 NEXT X
460 PRINT:PRINT
470 PRINT"ENTER FIRST START CO-ORDINATE
(LESS      THAN 10)"
480 INPUT P
490 IF P<1 OR P>10 THEN 480
500 PRINT"ENTER SECOND START CO-ORDINATE
 (LESS     THAN 10)"
510 INPUT Q
520 IF Q<1 OR Q>10 THEN 510
530 PRINT:PRINT
540 PRINT"ENTER FIRST END CO-ORDINATE (L
ESS        THAN 10)"
550 INPUT EP
560 IF EP<1 OR EP>10 THEN 550
570 PRINT"ENTER SECOND END CO-ORDINATE (
LESS       THAN 10)"
580 INPUT EQ
590 IF EQ<1 OR EQ>10 THEN 580
600 A$(P,Q)="O"
610 A$(EP,EQ)="X"
620 RETURN
```

Simultaneous Equations

```
10 REM SIMULTANEOUS EQUATIONS
20 POKE 53280,0:POKE 53280,0:PRINT"
{CLR}"
30 J=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
40 HS=0
50 FD=RND(0)
60 PRINT:PRINT"DECAY FACTOR IS"FD
70 GOSUB 550
80 PRINT"
{CLR}"
90 PRINT:PRINT
100 PRINT"ENTER NUMBER OF CELL X TO
          START (LESS THAN 40)"
110 INPUT CP:IF CP<1 OR CP>39 THEN 110
120 PRINT:PRINT
130 PRINT"WE HAVE"CP" X CELLS"
140 PRINT:PRINT
150 PRINT"ENTER NUMBER OF CELL Y TO
          START (LESS THAN 40)"
160 INPUT EP:IF EP<1 OR EP>39 THEN 160
170 PRINT "
{CLR}":PRINT:PRINT"PLEASE STAND BY..."
180 GOSUB 550:PRINT"
{CLR}"
190 DA=1
200 IF CP>EP/FD THEN CP=EP/FD
210 PRINT"---------------"
215 IF INT(CP)<0 THEN CP=0
216 IF INT(EP)<0 THEN EP=0
220 PRINT "TIME ELAPSED:"DA
230 PRINT INT(CP)"CELL X"
240 PRINT INT(EP)"CELL Y"
250 REM **************************
260 REM MAJOR CYCLE
270 GOSUB 550
280 DA=DA+1
290 PRINT"---------------"
```

```
300 PRINT "TIME ELAPSED:"DA
310 IF CP>EP/FD THEN CP=EP/FD
320 REM EQUATIONS FOLLOW; MODIFY PARTS
      OF THEM TO SEE WHAT HAPPENS
330 CP=CP+((8*CP-CP*EP/3)*FD)
340 EP=EP+((8*EP-EP*CP/3)*.01)
350 PRINT INT(CP)"CELL X"
360 PRINT INT(EP)"CELL Y"
370 IF EP<2 OR CP<2 THEN 410
380 GOSUB 550
390 GOTO 280
400 REM ************************
410 IF DA>HS THEN HS=DA
420 PRINT:PRINT
430 PRINT"YOUR CELL CLASH SIMULATION SUR
VIVED"
440 PRINT"FOR"DA"TIME PERIODS."
450 PRINT"-----------------------------
----"
460 PRINT"THE BEST SURVIVAL TIME SO FAR
IS"HS
470 GOSUB 550
480 PRINT"-----------------------------
----"
490 PRINT"DO YOU WANT A NEW RUN (Y OR N)
?"
500 GET A$
510 IF A$<>"Y" AND A$<>"N" THEN 500
520 IF A$="Y" THEN PRINT "
{CLR}":GOTO 60
530 PRINT"OK":PRINT:PRINT:END
540 REM ************************
550 FOR J=1 TO 2000:NEXT
560 RETURN
```

Life

```
10 REM CONWAY'S LIFE SIMULATION
20 REM DEFINED INITIAL COLONY
30 GOSUB 460:REM INITIALISE
40 REM ************************
50 REM MAJOR CYCLE
60 GENERATIN=GENERATIN+1
70 GOSUB 290:REM PRINTOUT
80 GOSUB110:REM EVOLVE
90 GOTO 60
100 REM ************************
110 REM EVOLVE
120 FOR X=2 TO 12
130 FOR Y=2 TO 12
140 C=0
150 IF A$(X-1,Y-1)="X" THEN C=C+1
160 IF A$(X-1,Y)="X" THEN C=C+1
170 IF A$(X-1,Y+1)="X" THEN C=C+1
180 IF A$(X,Y-1)="X" THEN C=C+1
190 IF A$(X,Y+1)="X" THEN C=C+1
200 IF A$(X+1,Y-1)="X" THEN C=C+1
210 IF A$(X+1,Y)="X" THEN C=C+1
220 IF A$(X+1,Y+1)="X" THEN C=C+1
230 IF A$(X,Y)="X" AND C<>2 AND C<>3 THE
N B$(X,Y)=CHR$(32)
240 IF A$(X,Y)=" " AND C=3 THEN B$(X,Y)=
"X"
250 NEXT Y
260 NEXT X
270 RETURN
280 REM ************************
290 REM PRINTOUT
300 PRINT"
{CLR}"
310 PRINT
320 PRINTTAB(4);"GENERATION"GENERATIN
330 PRINT
340 FOR X=2 TO 12
350 FOR Y=2 TO 12
```

```
360 A$(X,Y)=B$(X,Y)
370 PRINT A$(X,Y);
380 NEXT Y
390 FOR Y=12 TO 2 STEP -1
400 PRINT A$(X,Y);
410 NEXT Y
420 PRINT
430 NEXT X
440 RETURN
450 REM **************************
460 REM INITIALISE
470 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}{PUR}"
480 D=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
500 DIM A$(13,13),B$(13,13)
510 PRINT:PRINT"  PLEASE STAND BY..."
520 FOR X=1 TO 13
530 PRINT 14-X;
540 FOR Y=1 TO 13
550 REM FILL ARRAY WITH BLANKS
560 A$(X,Y)=CHR$(32)
570 B$(X,Y)=A$(X,Y)
580 NEXT Y
590 NEXT X
600 READ D:IF D=99 THEN 630
610 READ E:A$(D,E)="X":B$(D,E)="X"
620 GOTO 600
630 GENERATIN=0
640 RETURN
650 DATA 5,5,5,9,6,6,6,8
660 DATA 7,7
670 DATA 8,6,8,8,9,5,9,9
700 DATA 99
```

Robot Logo

```
100 REM ROBOT LOGO
110 GOSUB 1730:REM INITIALISE
120 GOTO 490
130 REM ****************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ****************************
340 REM INT UX,AX
350 UX=INT(UX+.5):AX=INT(AX+.5)
360 RETURN
370 REM ****************************
380 REM          PRINT OUT
390 PRINT "
{CLR}"
400 PRINT"STEP"PSN" > ";A$(PSN):PRINT
410 FOR J=1 TO DEPTH
420 FOR K=1 TO BREATH
430 PRINT Z$(J,K);
440 NEXT K
450 PRINT
460 NEXT J
```

```
470 RETURN
480 REM ****************************
490 REM READ PROGRAM
500 COUNT=COUNT+1
510 READ A$(COUNT)
520 IF A$(COUNT)="*" THEN 550
530 IF COUNT<20 THEN 500
540 REM ****************************
550 REM EXECUTE PROGRAM
560 PSN=0:REM PROGRAM STEP NUMBER
570 PSN=PSN+1
580 IF PSN=21 THEN 580:REM END
590 FLAG=0
600 M$=A$(PSN)
610 IF M$="*" THEN 610:REM END
620 N$=LEFT$(M$,2)
630 IF N$="ST" THEN 560:REM START AGAIN
640 IF N$="PR" THEN GOSUB 380:REM PRINTO
UT
650 IF N$="FO" THEN FLAG=1
660 IF N$="BA" THEN FLAG=2
670 IF N$="TU" THEN FLAG=3
680 IF N$="HO" THEN FLAG=4
690 IF N$="CL" THEN FLAG=5
700 IF N$="GO" THEN FLAG=6
710 IF N$="RA" THEN FLAG=7
720 IF N$="RE" THEN FLAG=8
730 IF N$="EN" THEN FLAG=9
740 IF N$="FA" THEN FLAG=10
750 ON FLAG GOSUB 780,940,1000,1160,1220
,1260,1400,1460,1530,1580
760 GOTO 570
770 REM ****************************
780 REM       FORWARD
790 M$=MID$(M$,4)
800 IF ASC(M$)=87 THEN M$=MID$(M$,6)
810 F$="F"
820 NUM=VAL(M$)
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN 880
850 IF AX<1 OR AX>BREATH THEN 880
```

```
860 Z$(UX,AX)=T$
880 IF F$="F" THEN UX=UX+UP:AX=AX+AC
890 IF F$="B" THEN UX=UX-UP:AX=AX-AC
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ****************************
940 REM            BACK
950 M$=MID$(M$,4)
960 IF ASC(M$)=78 THEN M$=MID$(M$,3)
970 F$="B"
980 GOTO 820
990 REM ****************************
1000 REM          TURN
1010 M$=MID$(M$,4)
1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
1030 NUM=VAL(M$)
1040 Y=INT((NUM+17.5)/45)
1050 IF Y=0 OR Y=8 THEN RETURN
1060 FOR J=1 TO Y
1070 IF UP=-1 AND AC=0 THEN AC=1:GOTO 11
30
1080 IF UP=0 AND AC=1 THEN UP=1:GOTO 113
0
1090 IF UP=1 AND AC=0 THEN AC=-1:GOTO 11
30
1100 IF UP=0 AND AC=-1 THEN UP=-1:GOTO 1
130
1110 IF UP=-1 AND AC=-1 OR UP=1 AND AC=1
 THEN AC=0:GOTO 1130
1120 IF UP=-1 AND AC=1 OR UP=1 AND AC=-1
 THEN UP=0
1130 NEXT J
1140 RETURN
1150 REM ****************************
1160 REM HOME
1170 AX=INT((BREATH+.5)/2)
1180 UX=INT((DEPTH+.5)/2)
1190 UP=-1:AC=0:REM FACES UP
1200 RETURN
1210 REM ****************************
```

```
1220 REM          CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ************************************
1260 REM          GO X,Y
1270 P=0
1280 P=P+1
1290 IF MID$(M$,P,1)="," THEN 1320
1300 IF P<LEN(M$) THEN 1280
1310 RETURN:REM ERROR
1320 UX=VAL(MID$(M$,4,P-1))
1330 AX=VAL(RIGHT$(M$,LEN(M$)-P))
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
1360 IF AC<1 OR AC>BREATH THEN 1380
1370 Z$(UX,AX)=R$
1380 RETURN
1390 REM ************************************
1400 REM          RANDOM
1410 AX=INT(RND(1)*BREATH)
1420 UX=INT(RND(1)*DEPTH)
1430 Z$(UX,AX)=R$
1440 RETURN
1450 REM ************************************
1460 REM          REPEAT
1470 M$=MID$(M$,4)
1480 IF ASC(M$)=69 THEN M$=MID$(M$,5)
1490 RECOUNT=VAL(M$)
1500 MARKER=PSN
1510 RETURN
1520 REM ************************************
1530 REM          END REPEAT
1540 RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ************************************
1580 REM          FACE
1590 M$=MID$(M$,4)
1600 IF ASC(M$)=69 THEN M$=MID$(M$,3)
1610 NUM=VAL(M$)
1620 Y=INT((NUM+17.5)/45)*45
```

```
1630 IF Y=0 OR Y=360 THEN UP=-1:AC=0
1640 IF Y=45 THEN UP=-1:AC=1
1650 IF Y=90 THEN UP=0:AC=1
1660 IF Y=135 THEN UP=1:AC=1
1670 IF Y=180 THEN UP=1:AC=0
1680 IF Y=225 THEN UP=1:AC=-1
1690 IF Y=270 THEN UP=0:AC=-1
1700 IF Y=315 THEN UP=-1:AC=-1
1710 RETURN
1720 REM **************************
1730 REM INITIALISE
1740 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}"
1750 J=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
1770 BREATH=40:REM CHARACTERS ACROSS
1780 DEPTH=24:REM CHARACTERS DOWN
1790 BREATH=BREATH-1
1800 DEPTH=DEPTH-3
1810 UP=-1:AC=0:REM STARTS FACING UP
1820 DIM A$(20):REM FOR ROBOT PROGRAM
1830 DIM Z$(DEPTH,BREATH):REM DISPLAY
1840 T$="X":REM PUT SYMBOL HERE YOU
     WANT TO USE FOR ROBOT'S TRAIL
1850 AX=0:UX=0
1860 REM FILL ARRAY WITH SPACES
1870 FOR J=1 TO DEPTH
1880 FOR K=1 TO BREATH
1890 Z$(J,K)=" "
1900 NEXT K
1910 NEXT J
1920 RETURN
```

Point Duty Robot

```
100 REM POINT-DUTY ROBOT
110 GOSUB 1730:REM INITIALISE
120 GOTO 490
130 REM ************************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ************************************
340 REM INT UX,AX
350 UX=INT(UX+.5):AX=INT(AX+.5)
360 RETURN
480 REM ************************************
490 REM READ PROGRAM
500 COUNT=COUNT+1
510 READ A$(COUNT)
520 IF A$(COUNT)="*" THEN 550
530 IF COUNT<20 THEN 500
540 REM ************************************
550 REM EXECUTE PROGRAM
560 PSN=0:REM PROGRAM STEP NUMBER
570 PSN=PSN+1
580 IF PSN=21 THEN 580:REM END
590 FLAG=0
```

```
600 M$=A$(PSN)
610 IF M$="*" THEN 610:REM END
620 N$=LEFT$(M$,2)
630 IF N$="ST" THEN 560:REM START AGAIN
650 IF N$="FO" THEN FLAG=1
660 IF N$="BA" THEN FLAG=2
670 IF N$="TU" THEN FLAG=3
680 IF N$="HO" THEN FLAG=4
700 IF N$="GO" THEN FLAG=5
710 IF N$="RA" THEN FLAG=6
720 IF N$="RE" THEN FLAG=7
730 IF N$="EN" THEN FLAG=8
740 IF N$="FA" THEN FLAG=9
750 ON FLAG GOSUB 780,940,1000,1160,1260
,1400,1460,1530,1580
760 GOTO 570
770 REM ****************************
780 REM        FORWARD
790 M$=MID$(M$,4)
800 IF ASC(M$)=87 THEN M$=MID$(M$,6)
810 F$="F"
820 NUM=VAL(M$)
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN 880
850 IF AX<1 OR AX>BREATH THEN 880
860 GOSUB 2000
880 IF F$="F" THEN UX=UX+UP:AX=AX+AC
890 IF F$="B" THEN UX=UX-UP:AX=AX-AC
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ****************************
940 REM              BACK
950 M$=MID$(M$,4)
960 IF ASC(M$)=75 THEN M$=MID$(M$,3)
970 F$="B"
980 GOTO 820
990 REM ****************************
1000 REM             TURN
1010 M$=MID$(M$,4)
1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
```

```
1030 NUM=VAL(M$)
1040 Y=INT((NUM+11.25)/22.5)
1050 IF Y=0 OR Y=16 THEN RETURN
1060 FOR J=1 TO Y
1065 IF UP=-2 AND AC=0 OR UP=2 AND AC=2
THEN AC=1:GOTO 1130
1070 IF UP=-2 AND AC=1 THEN AC=2:GOTO 11
30
1075 IF UP=-2 AND AC=2 OR UP=0 AND AC=-2
 THEN UP=-1:GOTO 1130
1080 IF UP=-1 AND AC=2 OR UP=1 AND AC=-2
 THEN UP=0:GOTO 1130
1085 IF UP=0 AND AC=2 OR UP=2 AND AC=-2
THEN UP=1:GOTO 1130
1090 IF UP=1 AND AC=2 THEN UP=2:GOTO 113
0
1095 IF UP=2 AND AC=1 THEN AC=0:GOTO 113
0
1100 IF UP=2 AND AC=0 THEN AC=-1:GOTO 11
30
1105 IF UP=2 AND AC=-1 THEN AC=-2:GOTO 1
130
1110 IF UP=-1 AND AC=-2 THEN UP=-2:GOTO
1130
1115 IF UP=-2 AND AC=-2 THEN AC=-1:GOTO
1130
1120 IF UP=-2 AND AC=-1 THEN AC=0
1130 NEXT J
1140 RETURN
1150 REM ************************
1160 REM          HOME
1170 AX=INT((BREATH+.5)/2)
1180 UX=INT((DEPTH+.5)/2)
1190 UP=-2:AC=0:REM FACES UP
1200 RETURN
1210 REM ************************
1220 REM          CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ************************
1260 REM          GO X,Y
```

```
1270 P=0
1280 P=P+1
1290 IF MID$(M$,P,1)="," THEN 1320
1300 IF P<LEN(M$) THEN 1280
1310 RETURN:REM ERROR
1320 UX=VAL(MID$(M$,4,P-1))
1330 AX=VAL(RIGHT$(M$,LEN(M$)-P))
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
1360 IF AC<1 OR AC>BREATH THEN 1380
1370 GOSUB 2000
1380 RETURN
1390 REM ************************
1400 REM       RANDOM
1410 AX=INT(RND(1)*BREATH)
1420 UX=INT(RND(1)*DEPTH)
1430 GOSUB 2000
1440 RETURN
1450 REM ************************
1460 REM       REPEAT
1470 M$=MID$(M$,4)
1480 IF ASC(M$)=69 THEN M$=MID$(M$,5)
1490 RECOUNT=VAL(M$)
1500 MARKER=PSN
1510 RETURN
1520 REM ************************
1530 REM     END REPEAT
1540 RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ************************
1580 REM       FACE
1590 M$=MID$(M$,4)
1600 IF ASC(M$)=69 THEN M$=MID$(M$,3)
1610 NUM=VAL(M$)
1620 Y=INT((NUM+17.5)/45)*45
1630 IF Y=0 OR Y=360 THEN UP=-2:AC=0
1635 IF Y=22.5 THEN UP=-2:AC=1
1640 IF Y=45 THEN UP=-2:AC=2
1645 IF Y=67.5 THEN UP=-1:AC=2
1650 IF Y=90 THEN UP=0:AC=2
```

```
1655 IF Y=112.5 THEN UP=1:AC=2
1660 IF Y=135 THEN UP=2:AC=2
1665 IF Y=157.5 THEN UP=2:AC=1
1670 IF Y=180 THEN UP=2:AC=0
1675 IF Y=202.5 THEN UP=2:AC=-1
1680 IF Y=225 THEN UP=2:AC=-2
1685 IF Y=247.5 THEN UP=1:AC=-2
1690 IF Y=270 THEN UP=0:AC=-2
1695 IF Y=292.5 THEN UP=-1:AC=-2
1700 IF Y=315 THEN UP=-2:AC=-2
1705 IF Y=337.5 THEN UP=-2:AC=-1
1710 RETURN
1720 REM ************************
1730 REM INITIALISE
1740 PRINT "
{CLR}"
1750 J=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
1770 BREATH=320:REM CHARACTERS ACROSS
1780 DEPTH=200:REM CHARACTERS DOWN
1790 BREATH=BREATH-1
1800 DEPTH=DEPTH-3
1810 UP=-2:AC=0:REM STARTS FACING UP
1820 DIM A$(20):REM FOR ROBOT PROGRAM
1830 REM TURN ON HIRES SCREEN
1840 BASE=2*4096:POKE 53272,PEEK(53272)O
R8
1850 POKE 53265,PEEK(53265)OR32
1860 REM SET SCREEN TO CYAN AND BLACK
1870 FOR I=BASE TO BASE+7999:POKE I,0:NE
XT I
1880 FOR I=1024 TO 2023:POKE I,3:NEXT I
1920 RETURN
2000 REM ****** SET A POINT *********
2010 ROW=INT(UX/8)
2020 CHAR=INT(AX/8)
2030 LINE=UXAND7
2040 BIT=7-(AXAND7)
2050 BYTE=BASE+ROW*320+CHAR*8+LINE
2060 POKE BYTE,PEEK(BYTE)OR2^BIT
2070 RETURN
```

Connect Four

```
10 REM CONNECT FOUR
20 REM A. W. PEARSON
30 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}{WHT}"
40 PRINT
50 PRINT
60 PRINT"CONNECT FOUR"
70 PRINT
80 PRINT "ENTER YOUR MOVE AS A NUMBER BE
TWEEN"
90 PRINT "1 AND 8, ENTER 0 FOR A NEW GAM
E"
100 FOR F=1 TO 1000:NEXT F
110 DIM A$(10,10),B(10,2)
120 FLAG=0
130 REM CHANGE NEXT LINE FOR YOUR OWN
      CHOICE OF SYMBOLS (C$-COMPUTER)
140 C$="M":H$="O":REM M FOR MACHINE!
150 FOR F=1 TO 8
160 B(F,1)=6
170 NEXT F
180 FOR F=1 TO 6
190 FOR G=1 TO 8
200 A$(F,G)="."
210 NEXT G
220 NEXT F
230 REM **************************
240 REM ACCEPT HUMAN MOVE
250 GOSUB 430
260 PRINT:PRINT"
{PUR}YOUR MOVE..."
270 INPUT A:PRINT"
{WHT}";
280 IF A=0 THEN RUN
290 IF A<1 OR A>9 THEN 270
300 L=0
310 IF A$(L+1,A)<>"." OR L=6 THEN 340
320 L=L+1
330 GOTO 310
```

```
340 IF L=0 THEN 270
350 A$(L,A)=H$
360 B(A,1)=B(A,1)-1
370 GOSUB 430
380 GOSUB 560
390 GOSUB 430
400 GOTO 260
410 REM ************************
420 REM PRINT BOARD
430 PRINT"
{CLR}"
440 FOR F=1 TO 6
450 FOR G=1 TO 8
460 PRINT A$(F,G);
470 NEXT G
480 PRINT
490 NEXT F
500 PRINT"12345678"
510 PRINT
520 IF FLAG=1 THEN PRINT"I HAVE WON":END
530 RETURN
540 REM ************************
550 REM COMPUTER MOVES
560 PRINT"MY MOVE..."
570 MV=0
580 FOR F=1 TO 8
590 B(F,2)=0
600 NEXT F
610 FOR F=1 TO 8
620 FOR X=-1 TO 1
630 FOR Y=-1 TO 1
640 IF B(F,1)=0 THEN 680
650 IF A$(B(F,1)+X,F+Y)="" OR A$(B(F,1)+
X,F+Y)="." THEN 680
660 IF A$(B(F,1)+X,F+Y)=H$ THEN GOSUB 81
0
670 IF A$(B(F,1)+X,F+Y)=C$ THEN GOSUB 91
0
680 NEXT Y
690 NEXT X
700 NEXT F
```

```
710 P=0
720 FOR F=1 TO 8
730 IF B(F,2)>P THEN P=B(F,2):N=F
740 NEXT F
750 A$(B(N,1),N)=C$
760 B(N,1)=B(N,1)-1
770 P=0
780 N=0
790 RETURN
800 REM ***************************
810 MV=2
820 M1=MV
830 IF A$(B(F,1)+(X*2),F+(Y*2))=H$ THEN
MV=MV+10
840 IF A$(B(F,1)-X,F-Y)=H$ THEN MV=MV+20
850 IF MV<>M1+10 THEN 870
860 IF A$(B(F,1)+(X*3),F+(Y*3))=H$ THEN
MV=MV+1000
870 B(F,2)=B(F,2)+MV
880 M1=0
890 RETURN
900 REM ***************************
910 MV=2
920 M1=MV
930 IF A$(B(F,1)+(X*2),F+(Y*2))=C$ THEN
MV=MV+9
940 IF A$(B(F,1)-X,F-Y)=C$ THEN MV=MV+20
950 IF MV<>M1+9 THEN 970
960 IF A$(B(F,1)+(X*3),F+(Y*3))=C$ THEN
MV=MV+2000:FLAG=1
970 B(F,2)=B(F,2)+MV
980 RETURN
```

Quevedo Chess Machine

```
10 REM QUEVEDO CHESS MACHINE
20 GOSUB 1510:REM INITIALISE
30 GOTO 60
40 GOSUB 1320:REM PRINT BOARD
50 GOSUB 110:REM COMPUTER MOVES
60 GOSUB 1320
70 GOSUB 1120:REM ACCEPT HUMAN MOVE
80 GOTO 40
90 END
100 REM ************
110 REM COMPUTER MOVES
120 IF QUIT=1 THEN 1080
130 W1=WK
140 REM ************
150 REM MOVE ONE
160 MOVE=1
170 KM=INT(BK/10)
180 RM=INT(R/10)
190 IF ABS(KM-RM)>3 THEN 330
200 A(R)=46
210 X=INT(BK/10):Y=INT(R/10)
220 IF X>Y THEN 270
230 IF A(R-10)<>46 THEN 270
240 IF A(R-19)=BK OR A(R-21)=BK OR A(R-2
0)=BK THEN 270
250 IF A(R-11)=BK OR A(R-9)=BK THEN 270
260 R=R-10:GOTO 300
270 IF A(R+10)<>46 THEN A(R)=R:GOTO 330
280 IF A(R+19)=BK OR A(R+21)=BK OR A(R+2
0)=BK THEN A(R)=R:GOTO 330
290 R=R+10
300 A(R)=ASC("R")
310 RETURN
320 REM ************
330 REM MOVE TWO
340 MOVE=2
350 KM=BK-10*KM
360 RM=R-10*RM
370 IF ABS(KM-RM)<2 THEN 480
```

```
380 A(R)=46
390 IF R>11 THEN IF (A(R-12)=BK OR A(R-2
)=BK OR A(R+8)=BK) THEN A(R)=R:GOTO 480
400 IF R>11 THEN IF (A(R-1)=BK OR A(R-11
)=BK OR A(R+9)=BK) THEN A(R)=R:GOTO 480
410 Y=BK-10*INT(BK/10)
420 Z=R-10*INT(R/10)
430 IF (Z=1 OR Y>Z) AND A(R+1)=46 THEN R
=R+1:GOTO 450
440 R=R-1
450 A(R)=ASC("R")
460 RETURN
470 REM **********
480 REM MOVE THREE
490 MOVE=3
500 WM=WK-10*INT(WK/10)
510 BM=BK-10*INT(BK/10)
520 IF ABS(WM-BM)<3 THEN 600
530 IF A(WK-1)<>46 OR A(WK-18)=BK OR A(W
K-2)=BK OR A(WK+8)=BK THEN 610
540 IF A(WK-11)=BK OR A(WK+9)=BK OR A(WK
-22)=BK THEN 610
550 A(WK)=46
560 WK=WK-1
570 A(WK)=ASC("K")
590 REM **********
600 REM MOVES FOUR, FIVE AND SIX
610 Z=ABS(INT(BK/10)-INT(WK/10))
620 IF Z=0 THEN 950
630 IF 2*INT(Z/2)=Z THEN 790
640 REM **********
650 REM MOVE FOUR
660 MOVE=4
670 A(R)=46
680 IF A(R-10)<>46 THEN 720
690 IF A(R-9)=BK OR A(R-11)=BK THEN 720
700 IF A(R-19)=BK OR A(R-21)=BK OR A(R-2
0)=BK THEN 720
710 R=R-10:GOTO 760
720 IF A(R+10)<>46 THEN A(R)=R:GOTO 790
730 IF A(R+19)=BK OR A(R+21)=BK OR A(R+2
```

```
Ø)=BK THEN A(R)=ASC("R"):GOTO 79Ø
74Ø IF A(R+11)=BK OR A(R+9)=BK THEN A(R)
=ASC("R"):GOTO 79Ø
75Ø R=R+1Ø
76Ø A(R)=ASC("R")
77Ø RETURN
78Ø REM ************
79Ø REM MOVE FIVE
8ØØ MOVE=5
81Ø J=INT(BK/1Ø)
82Ø K=BK-1Ø*J
83Ø L=INT(WK/1Ø)
84Ø M=WK-1Ø*L
85Ø Z=1Ø:IF J<L THEN Z=-1Ø
86Ø X=1:IF K<M THEN X=-1
87Ø A(WK)=46
88Ø W1=WK
89Ø WK=WK+Z+X
9ØØ G=ABS(WK-BK)
91Ø IF G=1 OR G=9 OR G=1Ø OR G=11 THEN W
K=W1:A(WK)=75:GOTO 95Ø
92Ø A(WK)=ASC("K")
93Ø RETURN
94Ø REM ************
95Ø REM MOVE SIX
96Ø MOVE=6
97Ø A(R)=46
98Ø IF R>11 THEN IF A(R-12)=BK OR A(R-2)
=BK OR A(R+8)=BK OR A(R-1)<>R THEN 1Ø7Ø
99Ø IF R>11 THEN IF (A(R-1)=BK OR A(R-11
)=BK OR A(R+9)=BK THEN 1Ø7Ø
1ØØØ Y=BK-1Ø*INT(BK/1Ø)
1Ø1Ø Z=R-1Ø*INT(R/1Ø)
1Ø2Ø IF (Z=1 OR Y>Z) AND A(R+1)=46 R=R+1
:GOTO 1Ø4Ø
1Ø3Ø R=R-1
1Ø4Ø A(R)=ASC("R")
1Ø5Ø RETURN
1Ø6Ø REM ************
1Ø7Ø GOSUB 132Ø
1Ø8Ø PRINT:PRINT
```

```
1090 PRINT"I CONCEDE TO THE MASTER"
1100 END
1110 REM **************
1120 REM ACCEPT HUMAN MOVE
1130 REM ENTER 'Q' TO QUIT
1140 MOVE=0
1150 PRINT">> MOVE TO (LETTER, NO.)";
1160 INPUT G$
1170 IF G$="Q" THEN 1280
1180 IF LEN(G$)<>2 THEN 1160
1190 Z=ASC(G$)
1200 IF Z<65 AND Z>72 THEN 1160
1210 X=VAL(RIGHT$(G$,1))
1220 IF X<1 OR X>8 THEN 1160
1230 A(BK)=46
1240 BK=10*(Z-64)+X
1250 IF A(BK)=ASC("R") THEN QUIT=1
1260 A(BK)=ASC("$")
1270 RETURN
1280 PRINT:PRINT
1290 PRINT"THANKS FOR THE GAME"
1300 END
1310 REM **************
1320 REM PRINT BOARD
1330 PRINT"
{CLR}"
1340 PRINT:PRINT
1350 IF MOVE>0 THEN PRINT"I USED MOVE"MO
VE
1360 IF MOVE=0 THEN PRINT
1370 PRINT:PRINT
1380 PRINT TAB(11);"ABCDEFGH"
1390 FOR J=8 TO 1 STEP -1
1400 PRINT TAB(8);J;
1410 FOR K=10 TO 80 STEP 10
1420 PRINT CHR$(A(J+K));
1430 NEXT K
1440 PRINT J
1450 NEXT J
1460 PRINT
1470 PRINT TAB(11);"ABCDEFGH"
```

```
148Ø PRINT:PRINT
149Ø RETURN
15ØØ REM **********
151Ø REM INITIALISATION
152Ø POKE 53280,Ø:POKE 53281,Ø:PRINT"
{CLR}"
153Ø J=RND(VAL(RIGHT$(TI$,2))):REM
 RANDOMIZE
154Ø MOVE=Ø
155Ø QUIT=Ø
156Ø DIM A(13Ø)
157Ø FOR J=1Ø TO 8Ø STEP 1Ø
158Ø FOR K=1 TO 8
159Ø A(J+K)=46:REM ASCII OF "."
16ØØ NEXT K
161Ø NEXT J
162Ø REM ** PLACE PIECES **
163Ø REM BLACK KING - HUMAN
164Ø BK=INT(RND(1)*3)+1
165Ø BK=1Ø*BK+BK+INT(RND(1)*5)
166Ø A(BK)=ASC("$")
167Ø REM WHITE KING - COMPUTER
168Ø WK=INT(RND(1)*4)+4
169Ø WK=1Ø*WK+WK+INT(RND(1)*2)
17ØØ IF WK=BK THEN 168Ø
171Ø A(WK)=ASC("K")
172Ø REM WHITE ROOK - COMPUTER
173Ø R=INT(RND(1)*4)+4
174Ø R=1Ø*R+R+INT(RND(1)*2)
175Ø IF R=BK OR R=WK THEN 173Ø
176Ø IF ABS(R-BK)<12 THEN 173Ø
177Ø A(R)=ASC("R")
178Ø RETURN
```

Washington D.C.

```
10 REM WASHINGTON D.C.
20 GOSUB 1160:REM INITIALISE
30 REM ************************
40 REM MAJOR CYCLE
50 P=INT(P+(P*273/ML))
60 GOSUB 160:REM PRINTOUT
70 GOSUB 510:REM CALCULATE
80 REM NOW CHECK END GAME
90 GOSUB 710:REM STANDARD OF LIVING
100 GOSUB 780:REM INFLATION RATE
110 GOSUB 840:REM UNEMPLOYMENT
120 IF GAME=1 THEN PRINT"
{CLR}":GOTO 890
130 GOTO 50
140 REM ************************
150 REM PRINTOUT
160 PRINT"
{CLR}{WHT}"
170 PRINT "PRESIDENT ";A$;":"
180 PRINT"YOUR ADMINISTRATION HAS BEEN I
N                    POWER FOR"Y+Z/4"YEARS"
190 PRINT"-------------------------------
----"
200 PRINT"-----STATE OF THE NATION------
----"
210 PRINT"-------------------------------
----"
220 PRINT"POPULATION"P
230 PRINT"NO. UNEMPLOYED"INT(U)"   "INT(1
00*U/P)"%"
240 PRINT"CURRENT WAGES $"WO" INFLATION"
INT(IP)"%"
250 PRINT"GOVT. EXPENDITURE LAST QTR. $M
"GE
260 PRINT "UNEMPLOYMENT COST $M"INT(10*G
U)/10
270 PRINT "INCOME FROM TAXES $M"INT(10*G
I)/10
```

```
280 PRINT"BUDGET SURPLUS(+)/DEFICIT(-) $
M"INT(BD*10)/10
290 PRINT"GROSS DOMESTIC PRODUCT $M"INT(
GDP*10)/10
300 IF Y+Z/4>.5 THEN PRINT"CHANGE IN LIV
ING STANDARD ";
305 IF Y+Z/4>.5 THEN PRINTINT((2*((RGDP/
AGDP)*100)-100)/3)"%"
310 PRINT"----------------------------
----"
320 PRINT"PUBLIC INVESTMENT"Y"Q"Z"$M"INT
(IV*10)/10
330 PRINT"----------------------------
----"
340 PRINT"OK, PRESIDENT ";A$;"..."
350 INPUT "ENTER GOVERNMENT SPENDING $M"
;GE
360 INPUT "
{PUR}ENTER COST OF WAGES $M";WN
370 PRINT"IS YOUR ADMINISTRATION IN FAVO
R OF"
380 PRINT"IMMIGRATION THIS QUARTER (Y/N)
?
{WHT}"
390 GET X$
400 IF X$<>"Y" AND X$<>"N" THEN 390
410 PRINT TAB(20);"OK...";X$
420 FOR H=1 TO 1000:NEXT H
430 IF X$<>"Y" THEN RETURN
440 PRINT "
{PUR}HOW MANY IMMIGRANTS WILL YOU ALLOW
                INTO THE US"
450 INPUT M:PRINT"
{PUR}";
460 IF M<0 THEN 450
470 P=P+M
480 RETURN
490 REM **************************
500 REM CALCULATIONS
510 CN=CN+(CN*IP/100)
520 U=P*(GE+IV)/(CN*10)+P*(IP/1000)
```

```
530 GU=U*WN/ML:REM UNEMPLOYMENT COST
540 GI=(((P-U)*WN*.4)/ML):REM INCOME
                              FROM TAXES
550 BD=BD+GI-GU-GE:REM BUDGET DEFICIT
560 AGDP=AGDP*(1+(IP/1000))
570 GDP=GE+IV+((P-U)*WN/ML)
580 RGDP=GDP*440/AGDP
590 IP=((GE+IV)/CN*.1+(WN/WO)/100)*100
600 IV=(CN*67)/(IP*IP)
610 WO=WN
620 Z=Z+1:IF Z>4 THEN Z=1:Y=Y+1
630 RETURN
640 REM ************************
650 REM CHECK BUDGET DEFICIT
660 IF BD>-1000 THEN RETURN
670 GAME=1
680 FLAG=1
690 RETURN
700 REM ************************
710 REM CHECK STANDARD OF LIVING
720 IF Y<.75 THEN RETURN
730 IF INT((2*((RGDP/AGDP)*100)-100)/3)>
-15 THEN RETURN
740 GAME=1
750 FLAG=2
760 RETURN
770 REM ************************
780 REM CHECK INFLATION RATE
790 IF IP<15 THEN RETURN
800 GAME=1
810 FLAG=3
820 RETURN
830 REM ************************
840 REM CHECK UNEMPLOYMENT
850 IF INT(U*100/P)<15 THEN RETURN
860 GAME=1
870 FLAG=4
880 RETURN
890 REM ************************
900 REM END OF THE GAME
910 PRINT"PRESIDENT ";A$;", YOUR"
```

```
920 PRINT"ADMINISTRATION'S POOR ECONOMIC
"
930 PRINT"PERFORMANCE HAS LED TO AN UNAC
CEPTABLE"
940 IF FLAG=1 THEN PRINT"BUDGET DEFICET"
950 IF FLAG=2 THEN PRINT"DROP IN THE STA
NDARD OF LIVING"
960 IF FLAG=3 THEN PRINT"RISE IN THE INF
LATION RATE"
970 IF FLAG=4 THEN PRINT"RISE IN UNEMPLO
YMENT"
980 PRINT"    AMONG OTHER THINGS..."
990 PRINT"-----------------------------
----"
1000 PRINT"THE LACK OF CONFIDENCE IN YOU
R"
1010 PRINT"ADMINISTRATION IS SO BAD THER
E ARE"
1020 PRINT"CALLS FOR YOU TO RESIGN...YOU
 STEP"
1030 PRINT"ASIDE TO ALLOW THE VICE-PRESI
DENT TO"
1040 PRINT"        OCCUPY THE OVAL OFFICE
"
1050 FOR H=1 TO 1000:NEXT H
1060 PRINT"-----------------------------
----"
1070 PRINT"YOU WERE PRESIDENT FOR"Y+(Z*.
25)"YEARS"
1080 PRINT"DURING YOUR TERM OF OFFICE, T
HE"
1090 PRINT"POPULATION ROSE BY"P-3*ML
1100 PRINT"THE UNEMPLOYED RATE BECAME"IN
T(U*1000/P)/10"%"
1110 PRINT"AND THE INFLATION RATE BECAME
"INT(IP*10)/10"%"
1120 PRINT"STANDARD OF LIVING CHANGED BY
"INT((2*((RGDP/AGDP)*100)-100)/3)"%"
1130 PRINT"AND THE BUDGET SURPLUS/DEFICI
T          WAS $M"INT(BD*10)/10
1140 END
```

```
1150 REM **************************
1160 REM INITIALISE
1170 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}"
1180 H=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
1190 ML=1000*1000
1200 P=3*ML:REM POPULATION
1210 U=P/10:REM UNEMPLOYMENT
1220 IV=236:REM INVESTMENT
1230 GE=118:REM GOVERNMENT EXPENDITURE
1240 GU=0:REM COST OF UNEMPLOYMENT
1250 GI=0:REM INCOME FROM TAXES
1260 WN=100:REM NEW WAGES
1270 WO=100:REM OLD WAGES
1280 IP=5:REM INFLATION PERCENT
1290 GDP=440:REM GROSS DOMESTIC PRODUCT
1300 AGDP=440:REM BASE YEAR GDP
1310 RGDP=440:REM REAL GDP
1320 CN=354:REM ECONOMIC CONSTANT
     (USED THROUGHOUT SIMULATION)
1330 Z=1:GAME=0:FLAG=0
1340 Y=0:REM YEAR
1350 PRINT "ENTER YOUR LAST NAME"
1360 INPUT A$
1370 RETURN
```

Stock Market

```
1Ø REM STOCK MARKET
2Ø POKE 53280,Ø:POKE 53281,Ø:PRINT"
{CLR}"
3Ø C=INT(RND(VAL(RIGHT$(TI$,2)))):REM
RANDOMIZE
4Ø DIM S(5),N(5),P(5),D(5)
5Ø S(1)=1.49:S(2)=1.99:S(3)=2.49:S(4)=2.
99:S(5)=3.49
6Ø N(1)=2ØØØ:N(2)=15ØØ:N(3)=12ØØ:N(4)=1Ø
ØØ:N(5)=8ØØ
7Ø BB=265:TV=15ØØØ:QQ=15ØØØ:DAY=1
8Ø PRINT:PRINT"
{PUR}ENTER YOUR GOAL FOR THIS SIMULATION
,"
9Ø PRINTTAB(8);"$16,ØØØ TO $1ØØ,ØØØ"
1ØØ INPUT GAL:REM INPUT GOAL
11Ø IF GAL<16ØØØ THEN PRINT "
{WHT}TOO LOW!":GOTO 8Ø
12Ø IF GAL>1ØØ*1ØØØ THEN PRINT "
{WHT}TOO HIGH!":GOTO 8Ø
13Ø REM **********************
14Ø REM MAJOR LOOP
15Ø FOR C=1 TO 5
16Ø REM ADJUST THE 55 IN NEXT LINE TO
MODIFY GAME; 8Ø VERY HARD, 3Ø VERY EASY
17Ø D(C)=INT(RND(1)*55)+1
18Ø P(C)=INT(RND(1)*(1ØØ-D(C)))+1
19Ø NEXT C
2ØØ GOSUB 23Ø
21Ø GOTO 46Ø
22Ø REM **********************
23Ø REM PRINTOUT
24Ø PRINT"
{CLR}{WHT}"
25Ø PRINT"----------------------------
----------"
26Ø PRINT "DAY"DAY"          YOUR GOAL IS $
"GAL
27Ø PRINT"----------------------------
----------"
```

```
280 PRINT"COMPANY NUMBER:"
290 PRINT TAB(2);1;TAB(9);2;TAB(16);3;TA
B(25);4;TAB(35);5
300 PRINT "CHANCE OF INCREASE (%):"
310 PRINT TAB(2);P(1);TAB(9);P(2);TAB(16
);P(3);TAB(25);P(4);TAB(35);P(5)
320 PRINT "CHANCE OF DECREASE (%):"
330 PRINT TAB(2);D(1);TAB(9);D(2);TAB(16
);D(3);TAB(25);D(4);TAB(35);D(5)
340 PRINT"CURRENT VALUE PER SHARE:"
350 PRINT "$";INT(S(1)*100)/100;TAB(8);"
$";INT(S(2)*100)/100;
360 PRINT TAB(15);"$";INT(S(3)*100)/100;
TAB(23);"$";INT(S(4)*100)/100;
370 PRINT TAB(30);"$";INT(S(4)*100)/100
380 PRINT"NO. OF SHARES HELD:"
390 PRINT TAB(2);N(1);TAB(9);N(2);TAB(16
);N(3);TAB(25);N(4);TAB(35);N(5)
400 PRINT"BANK $"INT(BB)" TOTAL WORTH $"
INT(100*TV)/100
410 PRINT"---------------------------
---------"
420 IF TV>GAL THEN PRINT"YOU'VE HIT YOUR
 FINANCIAL GOAL!":END
430 RETURN
440 REM **********************
450 REM        ** SELL **
460 PRINT"
{PUR}DO YOU WANT TO SELL ANY SHARES (Y/N
)?"
470 GET A$
480 IF A$<>"Y" AND A$<>"N" THEN 470
490 IF A$="N" THEN 690
500 GOSUB 230
510 PRINT"
{WHT}WHICH ONES TO SELL?";
520 GET A$
530 IF A$<"1" OR A$>"5" THEN 520
540 C=VAL(A$)
550 PRINT"
{WHT}     OK"C
```

```
560 PRINT"
{PUR}HOW MANY OF"C"TO SELL";
570 INPUT N
580 IF N>N(C) THEN PRINT"
{WHT}YOU DON'T HAVE THAT MANY!":GOTO 570
590 REM **********************
600 REM ADJUST FIGURES AFTER SALE
610 BB=BB+S(C)*N:REM ADD VALUE TO BANK
620 N(C)=N(C)-N:REM SUBTRACT NO. SOLD
630 TV=0:REM SET TOTAL WORTH TO ZERO
640 REM NOW DETERMINE CURRENT WORTH
650 FOR C=1 TO 5
660 TV=TV+N(C)*S(C)
670 NEXT C
680 TV=TV+BB:REM ADD IN BANK BALANCE
690 GOSUB 230
700 REM **********************
710 REM        ** BUY **
720 PRINT "
{PUR}DO YOU WANT TO BUY ANY SHARES (Y/N)
?"
730 GET A$
740 IF A$<>"Y" AND A$<>"N" THEN 730
750 IF A$="N" THEN 890
760 GOSUB 230
770 PRINT"
{PUR}WHICH COMPANY TO BUY?";
780 GET A$
790 IF A$<"1" OR A$>"5" THEN 780
800 C=VAL(A$)
810 PRINT"
{WHT}     OK"C
820 PRINT"
{PUR}HOW MANY OF"C"TO BUY";
830 INPUT N
840 IF N*S(C)>BB THEN PRINT"
{WHT}YOU DON'T HAVE ENOUGH MONEY!":GOTO
830
850 REM **********************
860 REM ADJUST FIGURES AFTER BUY
870 BB=BB-S(C)*N
```

```
880 N(C)=N(C)+N
890 TV=0
900 FOR C=1 TO 5
910 TV=TV+N(C)*S(C)
920 NEXT C
930 TV=TV+BB
940 GOSUB 230
950 REM **********************
960 REM MODIFY ALL INDICATORS
970 TV=0
980 FOR C=1 TO 5
990 K=INT(RND(1)*100)+1
1000 IF K<P(C) THEN S(C)=S(C)*(1+(P(C)/1
000))
1010 K=INT(RND(1)*100)+1
1020 IF K<D(C) THEN S(C)=S(C)/(1+(P(C)/1
000))
1030 TV=TV+(S(C)*N(C))
1040 NEXT C
1050 TV=TV+BB
1060 QQ=QQ+1
1070 W=(TV*100/QQ)-100
1080 IF W=0 THEN W=.1
1090 W=W+6
1100 IF W<1 THEN W=1
1110 IF W>15 THEN W=15
1120 RESTORE
1130 FOR T=1 TO W
1140 READ A$
1150 NEXT T
1160 PRINT
1170 REM **********************
1180 REM GIVE RATING, START NEW ROUND
1190 PRINT"
{WHT}YOUR RATING AFTER THAT ROUND OF"
1200 PRINT"TRADING IS '";A$;"'"
1210 PRINT:PRINT "
{PUR}    <PRESS SPACEBAR TO CONTINUE  >"
1220 GET ZZ$:IF ZZ$<>" " THEN 1220
1230 DAY=DAY+1
1240 GOTO 150
```

```
1250 DATA "HOPELESS","VERY, VERY POOR"
1260 DATA "TERRIBLE","AWFUL","BAD"
1270 DATA "VERY ORDINARY","AVERAGE"
1280 DATA "REASONABLE","A LITTLE ABOVE A
VERAGE"
1290 DATA "FAIRLY GOOD","GOOD","VERY GOO
D"
1300 DATA "GREAT","EXCELLENT","SUPERLATI
VE"
```

Detroit City

```
1Ø REM DETROIT CITY
2Ø GOSUB 1640:REM INITIALISE
3Ø GOTO 11Ø
4Ø MT=MT+1
5Ø GOSUB 65Ø
6Ø IF TP>2ØØ THEN 156Ø
7Ø PRINT"
{CYN}DO YOU WANT TO RESIGN (Y/N)?{WHT}"
8Ø GOSUB 1Ø1Ø
9Ø IF A$="Y" THEN PRINT"OK, CHIEF":END
1ØØ GOSUB 138Ø
11Ø GOSUB 65Ø
12Ø FOR T=1 TO 1ØØØ:NEXT T
13Ø GOSUB 85Ø
14Ø PRINT"DO YOU WANT TO EXPAND OUTPUT (
Y/N)?"
15Ø GOSUB 1Ø1Ø
16Ø IF A$="Y" THEN 1Ø8Ø
17Ø IF SF=1 THEN 21Ø
18Ø PRINT"DO YOU WANT TO SELL FACTORY 4
(Y/N)?"
19Ø GOSUB 1Ø1Ø
2ØØ IF A$="Y" THEN 125Ø
21Ø GOSUB 65Ø
22Ø INPUT "HOW MANY EMPLOYEES TO HIRE";H
E
23Ø NE=NE+HE:IF HE>Ø THEN 26Ø
24Ø INPUT "HOW MANY EMPLOYEES TO FIRE";H
E
25Ø NE=NE-HE
26Ø GOSUB 65Ø
27Ø P1=AS:REM SET P1 EQUAL TO OLD PRICE
28Ø INPUT "WHAT IS YOUR SELLING PRICE";A
S
29Ø REM  NEXT LINE REJECTS TOO BIG A
          CHANGE IN SELLING PRICE
3ØØ IF ABS(P1-AS)>25ØØ THEN PRINT"TOO BI
G A CHANGE FOR THE MARKET":GOTO 28Ø
```

```
310 PRINT"
{CLR}"
320 PRINT:PRINT:PRINT
330 MI=INT(RND(1)*4000)+48*1000:REM THIS
       MONTH'S SALES BY INDUSTRY
340 C=C+1:REM COUNTS NUMBER OF MONTHS
350 IF C<3 THEN 470
360 M=INT(RND(1)*10+1)/4:REM INFLATION
370 PRINT"
{CLR}"
380 PRINT"INFLATION RATE THIS QUARTER IS
"M"%"
390 PRINT"AVERAGE WAGES BILL WILL NOW RI
SE TO"
400 AW=(AW*M/100)+AW
410 PRINTTAB(8);"$"INT(AW)"  PER EMPLOYE
E"
420 GET ZZ$:IF ZZ$<>"" THEN 420
430 PRINT:PRINTTAB(12);"ANY KEY TO CONTI
NUE"
440 GET ZZ$:IF ZZ$="" THEN 440
450 FA=(FA*M/100)+FA
460 C=0
470 Y(1)=NE*15/12:REM SALES BASED ON
                NUMBER OF EMPLOYEES
480 Y(2)=(100-AS/FA)*MI/100:REM SALES
 BASED ON MONTHLY INDUSTRY SALES
490 REM NEXT LINE SETS LOWEST FIGURE
    FROM Y(1), Y(2), M(5) EQUAL TO Y(3)
500 IF Y(1)<Y(2) AND Y(1)<M(5) THEN Y(3)
=Y(1):GOTO 540
510 IF Y(2)<Y(1) AND Y(2)<M(5) THEN Y(3)
=Y(2):GOTO 540
520 Y(3)=M(5)
530 REM NEXT LINES DETERMINE
          MONTHLY SALES
540 IF ABS(P1-AS)<501 THEN Y(3)=3.6*Y(3)
/3
550 IF Y(3)>M(5) THEN Y(3)=Y(3)-1975:GOT
O 550
560 MC=(MC*M/100)+MC
```

```
570 EF=Y(3)/M(5)*100:REM EFFECIENCY %
 AS SALES DIVIDED BY TOTAL OUTPUT
580 AC=(MC*(ABS(85-EF)/3)/100)+MC:REM
        AVERAGE COST PER VEHICLE
590 MP=((Y(3)*(AS-AC))-(NE*AW/12)):REM
        MONTHLY PROFIT
600 MP=INT(MP/(100*1000))
610 TP=TP+MP/10:REM TOTAL PROFIT
                    IN MILLIONS
620 M=0
630 GOTO 40
640 REM **************************
650 REM REPORT PRINTOUT
660 PRINT"
{CLR}"
670 PRINT"INDUSTRY SALES"MI"IN MONTH"MT
680 IF MT>0 THEN PRINT"YOUR SALES:"INT(Y
(3))" ("INT(Y(3)*1000/MI)/10"% OF TOTAL
)"
690 PRINT"-------------------------
---"
700 PRINT"YOU HAVE"NE"EMPLOYEES"
710 PRINT"AVERAGE WAGES ARE $"AW
720 PRINT " OR $M"INT(AW*NE/(100*1000)/1
2)/10"PER MONTH"
730 PRINT"-------------------------
---"
740 IF MT=0 THEN RETURN
750 PRINT"AVERAGE COST PER VEHICLE IS $"
INT(AC)
760 PRINT"AND AVERAGE SELLING PRICE IS $
"INT(AS)
770 PRINT"SO THE AVERAGE PROFIT IS $"INT
(AS-AC)
780 PRINT"OR $M"INT((AS-AC)*Y(3)/(100*10
00))/10"PER MONTH"
790 PRINT"-------------------------
---"
800 PRINT"PROFIT FOR THE MONTH IS $M"MP/
10
810 PRINT"& TOTAL PROFIT TO DATE IS $M"I
NT(TP*10)/10
```

```
820 PRINT"-----------------------------
---"
830 RETURN
840 REM ***************************
850 REM MONTH REPORT
860 PRINT"
{CLR}"
870 IF MT>0 THEN PRINT"YOUR MONTHLY SALE
S ARE"INT(Y(3))
880 PRINT"-----------------------------
---"
890 PRINT"MAXIMUM MONTHLY OUTPUT:"
900 PRINT TAB(3);"FACTORY 1:"INT(M(1))
910 PRINT TAB(3);"FACTORY 2:"INT(M(2))
920 PRINT TAB(3);"FACTORY 3:"INT(M(3))
930 IF SF=1 THEN 960
940 PRINT TAB(3);"FACTORY 4:"INT(M(4))
950 PRINT"-----------------------------
---"
960 PRINT "TOTAL OUTPUT IS"INT(M(5))
970 PRINT"-----------------------------
---"
980 PRINT "EFFICIENCY LEVEL IS"INT(EF)"%
"
990 RETURN
1000 REM ***************************
1010 REM GET REPLIES
1020 GET A$
1030 IF A$<>"Y" AND A$<>"N" THEN 1020
1040 PRINT TAB(22);A$
1050 FOR J=1 TO 500:NEXT J
1060 RETURN
1070 REM ***************************
1080 REM INCREASE OUTPUT?
1090 IF M(4)=0 THEN X=15:GOTO 1110
1100 X=18
1110 PRINT"IT WILL COST $M"X" TO EXPAND"
1120 PRINTTAB(8);"OUTPUT BY 1%"
1130 PRINT"-----------------------------
----"
1140 PRINT"HOW MANY % DO YOU WISH TO RAI
SE OUTPUT"
```

```
115Ø INPUT EP:IF EP<Ø OREP>1ØØ THEN 115Ø
116Ø M(5)=Ø
117Ø FOR T=1 TO 4
118Ø M(T)=M(T)+M(T)*EP/1ØØ
119Ø M(5)=M(5)+M(T)
12ØØ NEXT T
121Ø TP=TP-EP*X
122Ø FOR T=1 TO 5ØØ:NEXT T
123Ø GOTO 17Ø
124Ø REM ************************
125Ø REM SALE OF FACTORY FOUR
126Ø PRINT "FACTORY FOUR IS VALUED FOR S
ALE AT $M1Ø4"
127Ø PRINT"YOU CAN'T REBUY IT LATER IF
          YOU SELL IT..."
128Ø PRINT"DO YOU WANT TO SELL (Y/N)?"
129Ø GOSUB 1Ø1Ø
13ØØ IF A$="N" THEN 21Ø
131Ø TP=TP+1Ø4
132Ø SF=1
133Ø M(5)=M(1)+M(2)+M(3)
134Ø M(4)=Ø
135Ø GOTO 17Ø
136Ø REM ************************
137Ø REM CHECK ON LOSSES
138Ø IF MP>Ø THEN SA=Ø:GOTO 148Ø
139Ø SA=SA+1
14ØØ IF SA>11 THEN 142Ø
141Ø GOTO 148Ø
142Ø PRINT"
{CLR}":PRINT
143Ø PRINT"YOU JUST MADE YOUR TWELTH MON
THLY"
144Ø PRINT"LOSS IN A ROW.............
..."
145Ø PRINTTAB(6);"YOUR EMPLOYMENT"
146Ø PRINTTAB(6);"IS HEREBY TERMINATED!!
"
147Ø END
148Ø IF TP>=-25Ø THEN 135Ø
149Ø PRINT"
{CLR}":PRINT
```

```
1500 PRINT"UNDER YOUR MANAGEMENT, THE CO
MPANY HAS"
1510 PRINT"LOST MORE THAN $M250!"
1520 GOTO 1450
1530 IF TP>200 THEN 1570
1540 RETURN
1550 REM **************************
1560 REM SWEET SWEET SUCCESS!!!
1570 PRINT"
{CLR}":PRINT
1580 PRINT"WELL DONE! THE COMPANY HAS MA
DE MORE"
1590 PRINT"   THAN $M200.   YOU'VE BEEN
MADE"
1600 PRINT"          A MEMBER OF THE BOARD
"
1610 FOR T=1 TO 2000:NEXT T
1620 END
1630 REM **************************
1640 REM INITIALISE
1660 T=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
1670 DIM M(5),Y(5)
1680 NE=12000:REM STARTING NO EMPLOYEES
1690 AW=22995:REM STARTING AVERAGE WAGE
1700 AC=11100:REM COST PRICE/VEHICLE
1710 AS=12000:REM SELLING PRICE
1720 MI=50*1000:MC=10100
1730 Y(3)=12500
1740 MS=25:EF=77:FA=160:SF=0:MT=0
1750 FOR J=1 TO 5
1760 READ M(J)
1770 NEXT J
1780 POKE 53280,0:POKE 53281,0:RETURN
1790 DATA 8900,3250,2500,1625,16275
```

Gridiron

```
5 POKE 53280,0:POKE 53281,0
10 REM GRIDIRON
20 PRINT"
{CLR}"
30 X=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
40 GOSUB 70
50 GOTO 220
60 REM ************************
70 FOR X=1 TO 1500:NEXT X
80 RETURN
90 REM ************************
100 PRINT A$;SA
110 PRINT B$;SB
120 RETURN
130 REM ************************
140 IF Z$=A$ THEN Z$=B$:RETURN
150 Z$=A$:RETURN
160 REM ************************
170 GET ZZ$:IF ZZ$<>"" THEN 170
180 PRINT"      > PRESS ANY KEY <"
190 GET ZZ$:IF ZZ$="" THEN 190
200 PRINT TAB(20);"OK":RETURN
210 REM ************************
220 REM INITIALISE
230 DEF FNA(X)=INT(RND(1)*X)+1
240 PRINT"
{WHT}ONE PLAYER OR TWO"
250 INPUT X
260 IF X<1 OR X>2 THEN 250
270 IF X=1 THEN VC=1:A$="SILICON COWBOYS
":GOTO 300
280 PRINT"WHAT IS THE NAME OF THE HOME T
EAM"
290 INPUT A$:IF A$="" THEN 290
300 PRINT"AND WHAT IS THE NAME OF THE VI
SITING     TEAM"
310 INPUT B$:IF B$="" THEN 310
```

```
315 A$="
{WHT}"+A$:B$="{PUR}"+B$
320 Z$=A$:NU=35
330 PRINT"
{CLR}"
340 PRINT"THERE ARE"INT(10*(60-(W/4)))/1
0"MINUTES TO GO"
350 PRINT TAB(8);Z$" TO KICK OFF"
360 PRINT"
{WHT}YOU ARE ON YOUR OWN"NU"YARD LINE"
370 IF VC=1 AND Z$=A$ THEN GOSUB 70:GOTO
 400
380 PRINT"TO KICK OFF..."
390 GOSUB 170
400 A=FNA(20)+40
410 PRINT Z$;" HAVE..."
420 FOR X=1 TO A
430 PRINT TAB(X/3);"KICKED"X"YARDS"
440 NEXT X
450 NU=NU+X
460 GOSUB 70
470 PRINT "
{PUR}THE BALL IS CAUGHT!"
480 GOSUB 70
490 A=FNA(30)+10
500 FOR X=1 TO A
510 PRINT TAB(X/5);"AND RETURNED"X"YARDS
"
520 NEXT X
530 NU=ABS(100-NU+X)
540 GOSUB 140
550 PRINT"----------------------------
----"
560 PRINT"THE BALL IS DOWN ON"
570 PRINT Z$;"'S"NU"YARD LINE"
580 IF Z$=A$ AND VC=1 THEN GOSUB 70:GOTO
600
590 GOSUB 170
600 TG=10:D=0:SL=NU
610 IF W=60 OR W=180 THEN 2010
620 IF W=120 THEN 2070
```

```
630 IF W=240 THEN 2140
640 PRINT"
{CLR}"
650 PRINTLEFT$(A$,6);SA;LEFT$(B$,6);SB
660 PRINT INT(10*(60-(W/4)))/10"MINUTES
TO GO"
670 GOSUB 70
680 PRINT"----------------------------
----"
690 PRINT Z$" IN POSSESSION"
700 PRINT TAB(4);D"DOWN"
710 PRINT TAB(4);TG"YARDS TO GO"
720 PRINT"----------------------------
----"
730 PRINT"START AT"SL"YARD LINE"
740 PRINT"NOW ON"NU"YARD LINE"
750 PRINT 100-NU"YARDS TO TOUCHDOWN"
760 PRINT"----------------------------
----"
770 PRINT "ON THIS PLAY ";
780 IF Z$=A$ THEN PRINT A$;" CAN":GOTO 8
00
790 PRINT B$;" CAN"
800 PRINT"EITHER 1 - THROW"
810 PRINT"        2 - CARRY"
820 PRINT"    OR 3 - PUNT"
830 P=0
840 IF A$=Z$ AND VC=1 AND D<3 THEN P=2:G
OTO 900
850 IF A$=Z$ AND VC=1 AND TG<7 THEN P=2:
GOTO 900
860 IF A$=Z$ AND VC=1 AND (100-NU)<31 TH
EN P=3:GOTO 900
870 IF A$=Z$ AND VC=1 THEN P=1:GOTO 900
880 GET K$:IF K$<"1" OR K$>"3" THEN 880
890 P=VAL(K$):PRINT TAB(10);"OK"P
900 GOSUB 70
910 W=W+1
920 PRINT"
{CLR}"
930 PRINT Z$;", YOUR QUARTERBACK HAS"
```

```
940 PRINTTAB(8);"GOT THE BALL"
950 PRINT"-----------------------------
----"
960 PRINT"WAIT FOR THE COUNT, ";Z$;","
970 PRINT TAB(8);"THEN HIT ANY KEY..."
980 GET ZZ$:IF ZZ$<>"" THEN 980
990 GOSUB 70
1000 GOSUB 2200
1010 IF E=11 THEN 2340
1020 PRINT"-----------------------------
-----"
1030 ON P GOTO 1050,1310,1590
1040 REM ************************
1050 PRINT"YOU'VE THROWN"E*5"YARDS"
1060 PRINT"AND THE PLAY IS..."
1070 A=FNA(8)
1080 IF A=1 THEN 1520
1090 A=FNA(E+1)
1100 IF A=1 THEN PRINTTAB(20);"...COMPLE
TE":GOTO 1220
1110 PRINT TAB(20);"...INCOMPLETE":D=D+1
1120 GOSUB 170
1130 PRINT"-----------------------------
-----"
1140 IF D>3 THEN 1160
1150 GOTO 610
1160 PRINT"THAT WAS YOUR 4TH DOWN"
1170 PRINT"AND YOU'VE LOST POSSESSION!!"
1180 D=0:TG=10:NU=ABS(100-NU):SL=NU
1190 GOSUB 70
1200 GOSUB 140
1210 GOTO 610
1220 GOSUB 170
1230 NU=NU+(E*5):TG=TG-(E*5)
1240 IF NU>100 THEN 1800
1250 IF TG<1 THEN 1280
1260 D=D+1:IF D>3 THEN 1160
1270 GOTO 610
1280 D=0:TG=10:SL=NU
1290 GOTO 610
1300 REM ************************
```

```
1310 A=FNA(15)
1320 IF A=1 THEN 1510
1330 E=A-5
1340 IF E<0 THEN 1440
1350 IF E=0 THEN E=1:GOTO 1370
1360 PRINT"GOOD SNAP, PASS AND RUN"
1370 PRINT"YOU'VE GAINED"E"YARDS"
1380 GOSUB 170
1390 TG=TG-E:NU=ABS(NU+E):D=D+1
1400 IF NU>100 THEN 1800
1410 IF TG<1 THEN 1280
1420 IF D>3 THEN 1160
1430 GOTO 610
1440 PRINT"GREAT RUNNING BY THE OPPOSITI
ON HAS"
1450 PRINT"CAUSED YOU TO LOSE"ABS(E)"YAR
DS"
1460 TG=TG-E:NU=NU+E:D=D+1
1470 GOSUB 170
1480 IF D>3 THEN 1160
1490 GOTO 610
1500 REM ************************
1510 PRINT"BAD SNAP...YOU'VE"
1520 PRINT"FUMBLED...AND"
1530 PRINT"YOU'VE LOST POSSESSION..."
1540 NU=100-NU:D=0:TG=10:SL=NU
1550 REM ************************
1560 GOSUB 170
1570 GOTO 460
1580 REM ************************
1590 PRINT"NICE PUNT..."
1600 PRINT"YOU'VE KICKED"E*4"YARDS"
1610 NU=NU+E*4
1620 IF NU>100 THEN 1650
1630 PRINT"-----------------------------
-----"
1640 GOTO 460
1650 A=FNA(3)
1660 IF A>1 THEN 1740
1670 PRINT"BUT YOU'VE MISSED THE GOAL!!"
1680 IF NU-E*4<80 THEN NU=ABS(100-(NU-E*
```

```
4)):GOTO 17ØØ
169Ø NU=2Ø
17ØØ D=Ø:TG=1Ø:SL=NU
171Ø GOSUB 14Ø
172Ø GOSUB 17Ø
173Ø GOTO 61Ø
174Ø PRINT".........AND SCORED!"
175Ø IF Z$=B$ THEN SB=SB+3:GOTO 177Ø
176Ø SA=SA+3
177Ø GOSUB 1ØØ
178Ø GOSUB 17Ø
179Ø NU=35:GOTO 33Ø
18ØØ PRINT"
{CLR}":GOSUB 4ØØØ
181Ø FOR X=1 TO 5
182Ø PRINTTAB(X*2);"TOUCHDOWN!!!"
183Ø NEXT X
184Ø IF Z$=A$ THEN SA=SA+6:GOTO 186Ø
185Ø SB=SB+6
186Ø GOSUB 1ØØ
187Ø PRINT"TO PLAY FOR EXTRA POINT"
188Ø GOSUB 17Ø
189Ø PRINT"-----------------------------
-----"
19ØØ PRINT "THE BALL IS SNAPPED...PREPAR
E TO KICK!"
191Ø GOSUB 7Ø
192Ø GOSUB 22ØØ
193Ø IF E>9 THEN PRINT"YOU MISSED":NU=2Ø
:GOTO 197Ø
194Ø PRINT"YOU SCORED...":NU=35
195Ø IF Z$=A$ THEN SA=SA+1:GOTO 198Ø
196Ø SB=SB+1:GOTO 198Ø
197Ø GOSUB 14Ø
198Ø GOSUB 1ØØ
199Ø GOSUB 17Ø
2ØØØ GOTO 33Ø
2Ø1Ø FOR X=1 TO 1Ø
2Ø2Ø PRINT TAB(2*X);"PERIOD OVER"
2Ø3Ø NEXT X
2Ø4Ø GOSUB 1ØØ
```

```
2050 GOSUB 170
2060 GOTO660
2070 FOR X=1 TO 10
2080 PRINT TAB(2*X);"HALF TIME"
2090 NEXT X
2100 GOSUB 100
2110 Z$=B$
2120 GOSUB 170
2130 NU=35:W=W+2:GOTO 330
2140 FOR X=1 TO 10
2150 PRINT TAB(2*X);"GAME OVER"
2160 NEXT X
2170 GOSUB 100
2180 END
2190 REM **************************
2200 E=0:X=10
2210 IF Z$=A$ AND VC=1 THEN PRINT"THIS O
NE FOR ";A$:GOTO2290
2220 E=E+1:X=X-1
2230 PRINT TAB(E);E
2240 FOR Y=1 TO X*1.5
2250 GET ZZ$:IF ZZ$<>"" THEN Y=X*1.5+1:R
ETURN
2260 NEXT Y
2270 IF E=11 THEN RETURN
2280 GOTO 2220
2290 FOR E=1 TO FNA(7)+2
2300 FOR J=1 TO 60:NEXT J
2310 PRINT TAB(E);E
2320 NEXT E
2330 RETURN
2340 PRINT"TOO LATE!"
2350 PRINT"YOU'VE BEEN SACKED!"
2360 E=FNA(4)
2370 IF E=3 THEN 2430
2380 PRINT"AND LOST FIVE YARDS!"
2390 TG=TG+5:D=D+1:NU=NU-5
2400 GOSUB 170
2410 IF D>3 THEN 1160
2420 GOTO 610
2430 PRINT"AND LOST POSSESSION!"
```

```
2440 D=0:NU=ABS(100-NU+5):SL=NU:TG=10
2450 GOSUB 170
2460 GOSUB 140
2470 GOTO 610
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP .7
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200
4120 POKE SID+1,L1
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -1
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

Tennis

```
10 REM TENNIS
20 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}"
30 D=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
40 AA=0:BB=0:T=0:KA=0
50 XA=0:YA=0:ZA=0
60 XB=0:YB=0:ZB=0
70 DEF FNA(X)=INT(RND(1)*X)+1
80 INPUT "ONE HUMAN PLAYER OR TWO";A
90 IF A<1 OR A>2 THEN 80
100 IF A=1 THEN A$="BJORNX":VC=1
110 IF VC=1 THEN 160
120 PRINT"PLEASE ENTER SIX-LETTER NAME"
130 INPUT "NAME OF FIRST PLAYER";A$
140 IF LEN(A$)<6 THEN A$=A$+CHR$(32):GOT
O 140
150 A$=LEFT$(A$,6)
160 INPUT "NAME OF SECOND PLAYER";B$
170 IF LEN(B$)<6 THEN B$=B$+CHR$(32):GOT
O 170
180 B$=LEFT$(B$,6)
190 S=1:AA=1:BB=1
200 PRINT"
{CLR}"
205 A$="
{PUR}"+A$:B$="{WHT}"+B$
210 P$=A$:R$=B$
220 REM **************************************
230 IF P$=A$ THEN R$=B$
240 IF P$=B$ THEN R$=A$
250 PRINTP$;" SERVING"
260 PRINT"DO YOU WANT TO SERVE 1 - FAST"
270 PRINT"                          OR 2 - SLOW"
280 IF P$=A$ AND VC=1 AND SC=0 THEN KB=1
:GOSUB 1720:GOTO 330
290 IF P$=A$ AND VC=1 AND SC=1 THEN KB=2
:GOSUB 1720:GOTO 330
```

```
300 GET K$
310 IF K$<"1" OR K$>"2" THEN 300
320 KB=VAL(K$)
330 PRINT:PRINTTAB(6);KB;TAB(10);"> IT'S
 A ";
340 IF KB=1 THEN PRINT"FAST";
350 IF KB=2 THEN PRINT"SLOW";
360 PRINT" SERVE..."
370 GOSUB 1720
380 IF KB=1 THEN EB=FNA(3):GOTO 400
390 EB=FNA(8)
400 IF EB=1 THEN 450
410 IF EB=3 AND SC=0 THEN 520
420 IF EB=3 AND SC=1 THEN 590
430 GOTO 670
440 REM ******************************
450 PRINT"
{CLR}":PRINT:GOSUB 4000
460 PRINT TAB(8);"....ACE...."
470 GOSUB 1720
480 SC=0
490 IF P$=A$ THEN 1140
500 GOTO 1150
510 REM ******************************
520 PRINT"
{CLR}":PRINT
530 PRINT TAB(12);"....OUT...."
540 PRINT TAB(8);"...SECOND SERVE..."
550 GOSUB 1720
560 SC=1
570 GOTO 230
580 REM ******************************
590 PRINT"
{CLR}":PRINT
600 PRINT TAB(12);"....OUT...."
610 PRINT TAB(8);"...DOUBLE FAULT..."
620 GOSUB 1720
630 SC=0
640 IF P$=A$ THEN 1150
650 GOTO 1140
660 REM ******************************
```

```
670 SC=0
680 PRINT"
{CLR}":PRINT
690 GET ZZ$:IF ZZ$<>"" THEN 690
700 PRINTR$;", THE BALL IS":PRINT"IN YOU
R COURT"
710 PRINT"------------------------"
720 IF R$=A$ AND VC=1 THEN 750
730 PRINT"HIT ANY KEY, WHEN YOU SEE THE
ZERO,              TO RETURN THE BALL..."
740 GET ZZ$:IF ZZ$<>"" THEN 740
750 X=4*FNA(3):Y=X
760 GOSUB 1720
770 E=5
780 PRINTTAB(2*(11-E));E
790 Y=Y-1
800 GET S$
810 IF S$<>"" AND E=0 THEN 890
820 IF S$<>"" THEN 990
830 IF Y>0 THEN 790
840 E=E-1:Y=X
850 IF E<-1 THEN 990
860 IF E=-1 AND R$=A$ AND VC=1 THEN 890
870 GOTO 780
880 IF KB=1 THEN EA=FNA(2):GOTO 1000
890 EA=FNA(4)
900 IF E=0 AND R$=A$ AND VC=1 THEN EA=FN
A(8)
910 IF EA=1 THEN 940
920 IF R$=A$ THEN R$=B$:GOTO 670
930 R$=A$:GOTO 670
940 PRINTR$;", YOU'VE HIT THE BALL"
950 PRINT TAB(8);"OUT OF PLAY..."
960 GOSUB 1720
970 IF R$=A$ THEN R$=B$:GOTO 1150
980 GOTO 1140
990 EA=FNA(4)
1000 IF EA=1 THEN 1070
1010 PRINT"YOU MISSED THE BALL, AND..."
1020 GOSUB 1720
1030 PRINT"   IT WAS IN...BAD MISTAKE"
```

```
1040 GOSUB 1720
1050 IF R$=A$ THEN R$=B$:GOTO 1140
1060 GOTO 1140
1070 PRINT"YOU MISSED THE BALL AND..."
1080 GOSUB 1720
1090 PRINT"IT WAS OUT...WELL LEFT"
1100 GOSUB 1720
1110 IF R$=A$ THEN R$=B$:GOTO 1140
1120 GOTO 1150
1130 REM ************************************
1140 AA=AA+1:GOTO 1160
1150 BB=BB+1
1160 IF AA<5 AND BB<5 THEN 1230
1170 IF (BB>4 AND AA<4) OR (BB>4 AND BB-
AA>1) THEN AA=1:BB=1:GOTO 1500
1180 IF (AA>4 AND BB<4) OR (AA>4 AND AA-
BB>1) THEN AA=1:BB=1:GOTO 1440
1190 IF AA>4 AND AA>BB THEN C$="ADV":D$=
"---":GOTO 1320
1200 IF BB>4 AND BB>AA THEN D$="ADV":C$=
"---":GOTO 1320
1210 C$="(DEUCE":D$="(DEUCE":GOTO 1320
1220 REM ************************************
1230 RESTORE
1240 FOR D=1 TO AA
1250 READ C$
1260 NEXT D
1270 RESTORE
1280 FOR D=1 TO BB
1290 READ D$
1300 NEXT D
1310 REM ************************************
1320 PRINT"
{CLR}"
1330 PRINT"----------------------"
1340 PRINT"           SET SET SET"
1350 PRINT"----------------------"
1360 PRINT"             1   2   3  GAME"
1370 PRINT A$;"   ";XA;"  ";YA;"  ";ZA;"  "
;C$
1380 PRINT B$;"   ";XB;"  ";YB;"  ";ZB;"  "
;D$
```

```
1390 PRINT"----------------------"
1400 GOSUB 1720
1410 IF T<>1 THEN 230
1420 END
1430 REM ************************************
1440 PRINT"
{CLR}"
1450 PRINT"GAME TO ";A$
1460 GOSUB 1720
1470 IF S=1 THEN XA=XA+1:C$="0":D$="0":G
OTO 1560
1480 IF S=2 THEN YA=YA+1:C$="0":D$="0":G
OTO 1580
1490 IF S=3 THEN ZA=ZA+1:C$="0":D$="0":G
OTO 1600
1500 PRINT"
{CLR}"
1510 PRINT"GAME TO ";B$
1520 GOSUB 1720
1530 IF S=1 THEN XB=XB+1:C$="0":D$="0":G
OTO 1560
1540 IF S=2 THEN YB=YB+1:C$="0":D$="0":G
OTO 1580
1550 IF S=3 THEN ZB=ZB+1:C$="0":D$="0":G
OTO 1600
1560 IF (XA>5 AND XB<5) OR (XA<5 AND XB>
5) THEN 1630
1570 IF (XA>5 AND XA-XB>1) OR (XA>5 AND
XB-XA>1) THEN 1630
1580 IF (YA>5 AND YB<5) OR (YA<5 AND YB>
5) THEN 1630
1590 IF (YA>5 AND YA-YB>1) OR (YA>5 AND
YB-YA>1) THEN 1630
1600 IF (ZA>5 AND ZB<5) OR (ZA<5 AND ZB>
5) THEN 1680
1610 IF (ZA>5 AND ZA-ZB>1) OR (ZA>5 AND
ZB-ZA>1) THEN 1680
1620 GOTO 1640
1630 S=S+1
1640 AA=1:BB=1
1650 IF P$=A$ THEN R$=A$:P$=B$:GOTO 1320
```

```
1660 P$=A$:R$=B$:GOTO 1320
1670 REM ******************************
1680 T=1
1690 GOTO 1320
1700 REM ******************************
1710 REM DELAY
1720 FOR M=1 TO 1000:NEXT M
1730 RETURN
1740 DATA "0","15","30","40"
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP .7
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200
4120 POKE SID+1,L1
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -1
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

Grand Prix

```
10 REM GRAND PRIX
20 GOSUB 2200:REM INITIALISE
30 GOSUB 1190:REM CHOOSE TRACK
40 REM ************************
50 REM MAJOR LOOP
60 GOSUB 120:REM PRINTOUT
70 GOSUB 280:REM ACCELERATION/CHECK
80 GOSUB 450:REM ENGINE/BRAKES
90 GOSUB 500:REM CORNER/POSITION
100 GOTO 60
110 REM ************************
120 REM PRINTOUT
130 PRINT"
{CLR}{PUR}"
140 PRINT"ENGINE TEMPERATURE"INT(ENG*10)
/10"C. (MAX. 200)"
150 PRINT"BRAKE TEMPERATURE:"INT(BRAK*10
)/10"C. (MAX. 500)"
160 PRINT" DISTANCE COVERED:"INT(DIST*10
)/10"METERS"
170 PRINT"                    :"INT(DIST*10
0/RR)/100"LAPS"
180 PRINT"YOU'RE IN POSITION"INT(FP)
190 PRINT"-----------------------------
------"
200 PRINT"    CURRENT SPEED:"INT(SPEED*1
0)/10"KPH"
210 PRINT"                    :"INT(SPEED*5
.555)/10"METERS PER MOVE"
220 PRINT"-----------------------------
------"
230 PRINT"CORNER APPROACHING"INT(APP)"ME
TERS"
240 PRINT"RECOMMENDED SPEED:"C(C)"KPH"
250 PRINT"-----------------------------
------"
260 RETURN
270 REM ************************
280 REM CHECK ACCELERATION AND FACTORS
```

```
290 GET X$:IF X$<>"Z" AND X$<>"M" AND X$
<>CHR$(32) THEN 290
300 PRINTTAB(12);"OK"
310 X=0
320 IF X$="M" THEN X=SPEED/15
330 IF X$="Z" THEN X=-SPEED/15
340 NUM=NUM+1:REM NUMBER OF MOVES
350 SPEED=SPEED+X
360 IF SPEED<0 THEN SPEED=0
370 TRAV=SPEED*.5555:REM DISTANCE
                          TRAVELLED
380 DIST=DIST+TRAV:REM TOTAL DISTANCE
                          TRAVELLED
390 ENG=ENG+(X/2)+.07:IF ENG<70 THEN ENG
=70+RND(1)*8:REM ENGINE TEMP
400 IF X>0 THEN BRAK=BRAK*.9:REM BRAKE
  TEMP FALLING; ACCELERATING
410 IF X<1 THEN BRAK=BRAK-(3*X)-RND(1)*3
:REM BRAKE TEMP INCREASING; BRAKING
420 IF BRAK<8 THEN BRAK=8+RND(1)*8
430 PRINT"
{WHT}";:RETURN
440 REM **********************
450 REM CHECK ENGINE/BRAKE TEMP
460 IF ENG>200 THEN PRINT"YOUR ENGINE HA
S OVER HEATED":GOTO 830
470 IF BRAK>500 THEN PRINT"YOUR BRAKES H
AVE OVERHEATED":GOTO 830
480 RETURN
490 REM **********************
500 REM CHECK CORNERING SPEED
            AND FIELD POSITION
510 APP=APP-TRAV
520 IF APP>0 THEN RETURN
530 CRASH=0
540 IF SPEED>(C(C)*1.125) THEN CRASH=1:G
OTO 690
550 IF SPEED>(C(C)*1.1) THEN GOTO 690
560 PNT=PNT+100-((C(C)*1.1)-SPEED):REM
CORNERING POINTS
570 NC=NC+1:REM NUMBER OF CORNERS
```

```
580 CP=96-(PNT/NC):REM CORNERING
                          POSITION
590 AM=AM+A(C):REM AVERAGE NUMBER
                     OF MOVES ALLOWED
600 RP=NUM-AM:REM RACING POSITION: YOUR
           MOVES MINUS AVERAGE MOVES
610 FP=(CP+RP)/2:REM FIELD POSITION IS
AVERAGE OF CORNER & RACE POSITIONS
620 IF FP<1 THEN FP=1
630 C=C+1
640 IF C=WW THEN C=1:REM LAP OVER
650 APP=APP+D(C)
660 IF LAP*QQ=AM THEN 910:REM RACE OVER
670 RETURN
680 REM **************************
690 REM CRASHED
700 PRINT"
{CLR}"
710 PRINT"YOU CORNERED AT"INT(10*SPEED)/
10"KPH"
720 PRINT"AND THE MAXIMUM SPEED WAS JUST
"C(C)
730 GOSUB 2330
740 PRINT"YOU SPIN OFF THE TRACK..."
750 GOSUB 2330
760 IF CRASH=1 THEN 830
770 PRINT"YOU'VE LOST 20 SECONDS, BUT YO
U'RE       ABLE TO REJOIN THE RACE"
780 NUM=NUM+10:SPEED=INT(2*C(C)/3)
790 PNT=PNT+50
800 GOSUB 2330
810 GOTO 570
820 REM **************************
830 PRINT".............AND CRASH!!!!!"
840 PRINT"-----------------------------
------"
850 PRINT"YOU ONLY COMPLETED"INT(DIST*10
)/10"METERS."
860 PRINT"OR"INT(DIST*100/RR)/100"LAPS A
ND AT THAT"
870 PRINT"STAGE YOU WERE IN POSITION"INT
(FP)
```

```
880 PRINT"----------------------------
------"
890 GOTO 1050
910 REM RACE OVER
920 PRINT"
{CLR}"
930 EFLAG=1
940 FOR X=1 TO 20
950 PRINTTAB(X);"WELL DONE, ";A$;"!!"
960 PRINTTAB(21-X);"WELL DONE, ";A$;"!!"
970 NEXT X
980 PRINT"----------------------------
------"
990 PRINT"YOU MANAGED TO LAST OUT THE FU
LL                  "LAP"LAP RACE..."
1000 PRINT"--------------------------
-------"
1010 GOSUB 2330
1020 PRINT"YOU FINISHED IN POSITION"INT(
FP)
1030 PRINT"AFTER STARTING IN 6TH POSITIO
N..."
1040 GOSUB 2330
1050 PRINT"YOUR AVERAGE SPEED WAS"INT(DI
ST*180/NUM)/100"KPH"
1060 GOSUB 2330
1070 IF RP<1 THEN RP=1
1080 IF CP<1 THEN CP=1
1090 PRINT"YOU WERE"INT(ABS(RP))"TH FAST
EST ON THE STRAIGHTS,"
1100 PRINT"AND"INT(ABS(CP))"TH FASTEST O
N THE CORNERS."
1110 PRINT:PRINT"PRESS 'S' FOR SAME RACE
, 'N' FOR NEW    RACE, 'E' TO END"
1120 GET I$:IF I$<>"S" AND I$<>"N" AND I
$<>"E" THEN 1120
1130 IF I$="E" THEN END
1140 GOSUB 2240
1150 RESTORE
1160 IF I$="S" THEN GOSUB 1490:LAP=L2AP:
GOTO 60
```

```
1170 IF I$="N" THEN PRINT"
{CLR}":GOSUB 1250:GOTO 60
1180 REM **********************
1190 REM NAME AND TRACK DATA
1200 INPUT "WHAT IS YOUR NAME, DRIVER";A
$
1210 PRINT
1220 FOR X=1 TO 3
1230 PRINT TAB(X*4);"OK, GOOD LUCK, ";A$
1240 GOSUB 2330:NEXT X
1250 PRINT"**********************************
*********"
1260 PRINT"WHICH RACE DO YOU WANT TO TAK
E PART IN:"
1270 PRINT
1280 PRINT TAB(7);"BRITISH GRAND PRIX 26
50MT   :1"
1290 PRINT TAB(7);"GERMAN  GRAND PRIX 17
00MT   :2"
1300 PRINT TAB(7);"ITALIAN GRAND PRIX 22
00MT   :3"
1310 PRINT TAB(7);"MONACO  GRAND PRIX 31
00MT   :4"
1320 PRINT
1330 PRINT TAB(7);"ENTER A NUMBER (1 TO
4)"
1340 GET K$
1350 IF K$<"1" OR K$>"4" THEN 1340
1360 GP=VAL(K$)
1370 PRINT"**********************************
**********"
1380 PRINT TAB(8);"OK, THE ";
1390 IF GP=1 THEN PRINT"BRITISH";
1400 IF GP=2 THEN PRINT"GERMAN";
1410 IF GP=3 THEN PRINT"ITALIAN";
1420 IF GP=4 THEN PRINT"MONACO";
1430 PRINT" RACE"
1440 PRINT"**********************************
*********"
1450 PRINT:PRINT"OVER HOW MANY LAPS?"
1460 INPUT LAP:IF LAP<1 THEN 1460
```

```
1470 LAP=INT(LAP+.5):L2AP=LAP
1480 REM **************************
1490 REM BRITISH DATA
1500 SPEED=140
1510 FOR X=1 TO 9
1520 READ D(X):REM DISTANCE BETWEEN
                              CORNERS
1530 NEXT X
1540 DATA 800,400,250,200,250,300,100,10
0,250
1550 FOR X=1 TO 9
1560 READ C(X)
1570 NEXT X
1580 DATA 150,90,175,200,200,90,90,150,1
50
1590 FOR X=1 TO 9
1600 READ A(X):REM AVERAGE NUMBER OF
       MOVES ALLOWED BETWEEN CORNERS
1610 NEXT X
1620 DATA 8,4,2,2,2,2,1,1,2
1630 APP=800:WW=10:QQ=24:RR=2650
1640 IF GP=1 THEN RETURN
1650 REM **************************
1660 REM GERMAN DATA
1670 SPEED=85
1680 FOR X=1 TO 7
1690 READ D(X)
1700 NEXT X
1710 DATA 600,200,100,150,250,200,200
1720 FOR X=1 TO 7
1730 READ C(X):REM RECOMMENDED
                    MAXIMUM CORNERING SPEED
1740 NEXT X
1750 DATA 90,175,120,90,200,200,175
1760 FOR X=1 TO 7
1770 READ A(X)
1780 NEXT X
1790 DATA 6,2,1,2,2,2,2
1800 APP=600:WW=8:QQ=17:RR=1700
1810 IF GP=2 THEN RETURN
1820 REM **************************
```

```
1830 REM ITALIAN DATA
1840 SPEED=108
1850 FOR X=1 TO 7
1860 READ D(X)
1870 NEXT X
1880 DATA 800,300,100,150,300,350,200
1890 FOR X=1 TO 7
1900 READ C(X):REM RECOMMENDED
                  MAXIMUM CORNERING SPEED
1910 NEXT X
1920 DATA 120,90,90,150,200,120,150
1930 FOR X=1 TO 7
1940 READ A(X)
1950 NEXT X
1960 DATA 8,3,3,1,3,3,2
1970 APP=800:WW=8:QQ=22:RR=2200
1980 IF GP=3 THEN RETURN
1990 REM **************************
2000 REM MONACO DATA
2010 SPEED=162.5
2020 FOR X=1 TO 14
2030 READ D(X)
2040 NEXT X
2050 DATA 400,100,100,300,400,300,150,20
0,200,200
2060 DATA 150,150,200,250
2070 FOR X=1 TO 14
2080 READ C(X):REM RECOMMENDED
                  MAXIMUM CORNERING SPEED
2090 NEXT X
2100 DATA 175,150,175,200,120,200,175,90
,175,150
2110 DATA 150,175,120,150
2120 FOR X=1 TO 14
2130 READ A(X)
2140 NEXT X
2150 DATA 4,1,1,3,4,3,1,2,2,2
2160 DATA 1,2,2,2
2170 APP=400:WW=15:QQ=30:RR=3100
2180 RETURN
2190 REM **************************
```

```
2200 REM INITIALISATION
2210 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}"
2220 X=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
2230 DIM A(14),C(14),D(14)
2240 C=1:FP=6:PNT=0:NC=0:CP=0
2250 AM=0:RP=0:APP=0
2260 NUM=0:REM NUMBER OF MOVES
2270 ENG=100:BRAK=10:TRAV=0:DIST=0
2280 EFLAG=0
2290 X=0
2300 RETURN
2310 REM **************************
2320 REM DELAY
2330 FOR O=1 TO 1000:NEXT O
2340 RETURN
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP .7
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200
4120 POKE SID+1,L1
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -1
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

Flight Simulation

```
10 REM FLIGHT SIMULATION
20 RPT=0
30 LD=INT(RND(1)*360)
40 DIM E$(1000):REM THIS HOLDS FLIGHT
                                 RECORD
50 DIM A$(7),C$(7):REM THESE ARRAYS
 HOLD HORIZON AND COMPASS OUTPUT
60 REM **************************
70 GOSUB 2320:REM INITALISE
80 IF CRASH=0 THEN GOSUB 820:REM HORIZON
                                /COMPASS
90 GOSUB 500:REM PRINTOUT
100 IF CRASH=1 THEN END
110 IF LND=1 AND UFLAG=1 THENGOSUB4000:P
RINT"WELL DONE. A PERFECT LANDING!":END
120 IF LND=1 AND UFLAG=0 THEN PRINT"YOUR
 WHEELS ARE UP":GOSUB 1780:GOTO 90
130 T=AIRSPEED:SALL=0
140 GET X$
150 IF X$="R" THEN RPT=1:GOTO 70
160 IF RPT=1 AND E$(CLOCK+1)="" THEN RPT
=0:GOTO 140
170 IF RPT=1 THEN X$=E$(CLOCK+1)
180 IF X$="" THEN 140
190 IF CLOCK<999 THEN E$(CLOCK+1)=X$
200 IF TAKEOV=1 THEN ELEVATE=INT(ELEVATE
+RND(1)*2-RND(1)*3)
210 IF AIRSPEED<3 THEN 290
220 IF X$="Q" THEN ELEVATE=ELEVATE+5:EFL
AG=5:IF ELEVATE>60 THEN SALL=1
230 IF X$="A" THEN ELEVATE=ELEVATE-5:EFL
AG=-5:IF ELEVATE<-70 THEN SALL=-1
240 IF SALL<>0 THEN GOSUB 1640
250 IF ALTIMETER<1 THEN 290:REM PREVENTS
      DRAMATIC TURNS ON THE GROUND
260 IF X$="Z" THEN WA=WA-.5:ANG=ANG-6:IF
 WA<-3 THEN WA=-3
270 IF X$="M" THEN WA=WA+.5:ANG=ANG+6:IF
 WA>3 THEN WA=3
```

```
280 ANG=INT(ANG+RND(1)*2-RND(1)*2)
290 IF X$=CHR$(32) THEN AIRSPEED=AIRSPEE
D+8.5
300 IF X$="." THEN AIRSPEED=AIRSPEED-7
310 AIRSPEED=AIRSPEED-ELEVATE/5
320 IF UFLAG=1 THEN AIRSPEED=AIRSPEED-1.
5:FUEL=FUEL-.5
330 IF AIRSPEED<0 THEN AIRSPEED=0
340 IF AIRSPEED>400 THEN AIRSPEED=400
350 IF X$="1" AND UFLAG=0 THEN UFLAG=1:G
OTO 370
360 IF X$="1" AND UFLAG=1 THEN UFLAG=0
370 FUEL=FUEL-(ABS(T-AIRSPEED)/10)-3.75
380 IF FUEL<1 THEN GOSUB 1780
390 IF TAKEOV=1 THEN 420
400 IF ELEVATE>10 AND AIRSPEED>45 AND AI
RSPEED<60 AND UFLAG=1 THEN TAKEOV=1
410 IF TAKEOV=0 THEN ALTIMETER=0:GOTO 45
0
420 IF LND=0 AND AIRSPEED<30 THEN ELEVAT
E=ELEVATE-5:ALTIMETER=9*ALTIMETER/10
430 ALTIMETER=ALTIMETER+INT(((ELEVATE+.1
)*AIRSPEED)+EFLAG*AIRSPEED/1000)/80
440 IF ALTIMETER<300 AND TAKEOV=1 THEN A
LTIMETER=ALTIMETER+AIRSPEED/30+ELEVATE
450 IF ALTIMETER<0 THEN GOSUB 1780:REM
      CRASH
460 REM CHANGE NEXT TWO LINES TO MAKE IT
      EASIER (OR EVEN HARDER) TO LAND
470 IF ALTIMETER>15 AND AIRSPEED>20 OR T
AKEOV=0 THEN 80
480 IF ABS(ANG-LD)<13 OR ABS(ANG+360-LD)
<13 THEN LND=1:GOTO 80
485 GOTO 80
490 REM **************************
500 REM PRINTOUT
510 PRINT"
{CLR}"
520 PRINT"    HORIZON";TAB(20);"HEADING"
530 EV=INT(ELEVATE/10)
540 IF EV>2 THEN EV=2
```

```
550 IF EV<-2 THEN EV=-2
560 IF EV<>0 AND TAKEOV=1 AND CRASH=0 TH
EN GOSUB 1920
570 PRINT ":--------------------:----------:"
580 FOR J=1 TO 7
590 PRINT":  ";A$(J);"  :";C$(J);"  :"
600 A$(J)=""
610 NEXT J
620 PRINT ":-------------------------------:"
630 DIST=DIST+ABS((COS(ELEVATE))*AIRSPEE
D)/360
640 CLOCK=CLOCK+1
650 PRINT":RANGE"INT(DIST*10)/10":  TIME"
INT(CLOCK)/10;":"LD
660 PRINT ":-------------------------------:"
670 PRINT":AIRSPEED  :"INT(AIRSPEED)
680 PRINT":";LEFT$(Q$,INT(AIRSPEED/20));
">"
690 PRINT":ALTIMETER:"INT(ALTIMETER);
700 IF ANG<0 THEN PRINTTAB(19);360+ANG"D
EG."
710 IF ANG>=0 THEN PRINTTAB(19);ANG"DEG.
"
720 MR=INT(ALTIMETER/30):IF MR>20 THEN M
R=20
730 PRINT":";LEFT$(Q$,MR);">"
740 PRINT":FUEL       :"INT(FUEL)
750 PRINT":";LEFT$(Q$,INT(FUEL/45));">"
760 PRINT ":-------------------------------:"
770 PRINT":ELEVATION:"ELEVATE":  ";:GOSUB
 2210:PRINTU$
780 IF UFLAG=1 THEN PRINT ":";TAB(5);">
UNDERCARRIAGE DOWN <  :"
790 IF UFLAG=0 THEN PRINT ":";TAB(6);">
UNDERCARRIAGE UP    <  :"
800 RETURN
810 REM ****************************
820 REM ASSIGN HORIZON/COMPASS
830 IF ABS(INT(WA+.5))=3 THEN GOSUB 980
840 IF ABS(INT(WA+.5))=2 THEN GOSUB 1070
850 IF ABS(INT(WA+.5))=1 THEN GOSUB 1160
```

```
860 IF INT(WA+.5)=0 THEN GOSUB 1250
870 REM NEXT TWO LINES USED TO
    GRADUALLY STRAIGHTEN UP WINGS
880 IF WA>0 THEN WA=WA-.2
890 IF WA<0 THEN WA=WA+.2
900 IF WA>.2 THEN 1350
910 FOR Z=1 TO 7
920 M$(8-Z)=A$(Z)
930 NEXT Z
940 FOR Z=1 TO 7
950 M$(Z)=A$(Z)
960 NEXT Z
970 GOTO 1350
980 REM WA=3 OR -3
990 A$(1)="                  *  "
1000 A$(2)="                **   "
1010 A$(3)="              **     "
1020 A$(4)="           **        "
1030 A$(5)="        **           "
1040 A$(6)="    **               "
1050 A$(7)="**                   "
1060 RETURN
1070 REM WA=2 OR -2
1080 A$(1)="                     "
1090 A$(2)="                   **  "
1100 A$(3)="               ***     "
1110 A$(4)="         ***           "
1120 A$(5)="    ***                "
1130 A$(6)="***                    "
1140 A$(7)="                       "
1150 RETURN
1160 REM WA=1 OR -1
1170 A$(1)="                       "
1180 A$(2)="                       "
1190 A$(3)="                  *****"
1200 A$(4)="      ******           "
1210 A$(5)="****                   "
1220 A$(6)="                       "
1230 A$(7)="                       "
1240 RETURN
1250 REM WA=0
```

```
1260 A$(1)="                              "
1270 A$(2)="                              "
1280 A$(3)="                              "
1290 A$(4)="**************************"
1300 A$(5)="                              "
1310 A$(6)="                              "
1320 A$(7)="                              "
1330 RETURN
1340 REM **************************
1350 REM ASSIGN COMPASS STRINGS
1360 F2=ANG-F1
1365 F2=ANG
1370 IF F2<0 THEN FA=INT((F2+375)/30)
1380 IF F2>=0 THEN FA=INT((F2+15)/30)
1390 IF FA=12 THEN FA=0
1400 C$(1)="    .N.   "
1410 IF FA=11 THEN C$(2)="   .@:..  ":GOTO
 1450
1420 IF FA=0 THEN C$(2)="   ..@..  ":GOTO
1450
1430 IF FA=1 THEN C$(2)="   ..:@.  ":GOTO
1450
1440 C$(2)="   ..:..  "
1450 IF FA=10 THEN C$(3)=" .@ :  .. ":GOTO
 1480
1460 IF FA=2 THEN C$(3)=" ..  :  @. ":GOTO
1480
1470 C$(3)=" ..  :  .. "
1480 IF FA=9 THEN C$(4)=" W@-X--E":GOTO
1510
1490 IF FA=3 THEN C$(4)=" W--X-@E":GOTO
1510
1500 C$(4)=" W--X--E"
1510 IF FA=8 THEN C$(5)=" .@ :  .. ":GOTO
1540
1520 IF FA=4 THEN C$(5)=" ..  :  @. ":GOTO
1540
1530 C$(5)=" ..  :  .. "
1540 IF FA=7 THEN C$(6)="   .@:..  ":GOTO
1580
1550 IF FA=6 THEN C$(6)="   ..@..  ":GOTO
```

```
1580
1560 IF FA=5 THEN C$(6)="   ..:@.  ":GOTO
1580
1570 C$(6)="   ..:..  "
1580 C$(7)="    .S.   "
1590 IF ANG>360 THEN ANG=ANG-360
1600 F2=ANG
1610 IF W>0 THEN W=W-.4
1620 IF W<0 THEN W=W+.4
1630 RETURN
1640 REM STALL/FALL
1650 IF SALL=-1 THEN 1710
1660 FOR J=1 TO 10
1670 PRINTTAB(J);"YOU HAVE STALLED!!"
1680 NEXT J
1690 AIRSPEED=AIRSPEED/4
1700 RETURN
1710 FOR J=1 TO 10
1720 PRINTTAB(J);"UNCONTROLLED DIVE!!"
1730 PRINTTAB(21-J);"PULL UP!!"
1740 NEXT J
1750 ALTIMETER=ALTIMETER/4
1760 RETURN
1770 REM ************************
1780 REM CRASH
1790 CRASH=1
1800 ALTIMETER=0
1810 M$="** *C R**  A ** S* H* !!*":REM
     25 CHARACTERS LONG
1820 FOR J=1 TO 20
1830 PRINTTAB(J);"CRASH!"
1840 PRINTTAB(21-J);"CRASH!"
1850 NEXT J
1860 FOR J=1 TO 7
1870 G=INT(RND(1)*11)+1
1880 A$(J)=MID$(M$,G,14)
1890 NEXT J
1900 RETURN
1910 REM ************************
1920 REM ADJUST HORIZON
1930 G$="                    ":REM 14 SPACES
```

```
1940 ON EV+3 GOSUB 1960,2020,2070,2080,2
140
1950 RETURN
1960 REM EV=-2
1970 FOR J=1 TO 4
1980 A$(J)=A$(J+3)
1990 NEXT J
2000 A$(5)=G$:A$(6)=G$:A$(7)=G$
2010 RETURN
2020 REM EV=-1
2030 FOR J=1 TO 6
2040 A$(J)=A$(J+1)
2050 NEXT J
2060 A$(7)=G$
2070 RETURN:REM EV=0
2080 REM EV=1
2090 FOR J=6 TO 1 STEP -1
2100 A$(J+1)=A$(J)
2110 NEXT J
2120 A$(1)=G$
2130 RETURN
2140 REM EV=2
2150 FOR J=4 TO 1 STEP -1
2160 A$(J+3)=A$(J)
2170 NEXT J
2180 A$(1)=G$:A$(2)=G$:A$(3)=G$
2190 RETURN
2200 REM **********************
2210 REM INPUT INTO COMMAND NAME
2220 U$="--------"
2230 IF X$=CHR$(32) THEN U$="THROTTLE ON
"
2240 IF X$="." THEN U$="THROTTLE OFF"
2250 IF X$="Q" AND ALTIMETER>0 THEN U$="
CLIMB"
2260 IF X$="Q" AND ALTIMETER=0 THEN U$="
NOSE UP"
2270 IF X$="A" THEN U$="NOSE DOWN"
2280 IF X$="Z" THEN U$="BANK LEFT"
2290 IF X$="M" THEN U$="BANK RIGHT"
2300 RETURN
```

```
2310 REM **************************
2320 REM INITIALIZATION
2330 POKE 53280,0:POKE 53281,0:PRINT"
{CLR}{PUR}"
2340 J=RND(VAL(RIGHT$(TI$,2))):REM
RANDOMIZE
2350 Q$="---------------------":REM
    21 CHARACTERS IN STRING
2360 UFLAG=1:REM UNDERCARRIAGE -
                  1 - DOWN, 0 - UP
2370 EFLAG=1:REM CLIMB RATE
2380 ANG=0:TAKEOV=0:LND=0
2390 AIRSPEED=0
2400 DIST=0:REM DISTANCE COVERED 'RANGE'
2410 ALTIMETER=0
2420 ELEVATE=0:REM ANGLE OF ELEVATION
2430 WA=0:REM WING ANGLE; USED IN
                       HORIZON PRINTOUT
2440 FUEL=750:CRASH=0:F2=0:F1=0:REM
 FOR DIRECTION CHANGE/COMPASS ROUTINE
2450 CLOCK=0:REM TIME
2460 X$=""
2470 RETURN
4000 SID=54272
4010 FOR L1=0 TO 23
4020 POKE SID+L1,0
4030 NEXT L1
4040 POKE SID+24,15
4050 POKE SID+5,15
4060 POKE SID+6,255
4070 POKE SID+4,17
4080 FOR L1=48 TO 220 STEP .7
4090 POKE SID+1,L1
4100 NEXT L1
4110 FOR L1=28 TO 200
4120 POKE SID+1,L1
4130 NEXT L1
4140 FOR L1=200 TO 28 STEP -1
4150 POKE SID+1,L1
4160 NEXT L1
4170 POKE SID+1,0
4180 RETURN
```

BBC Micro Listings

Space Landing

```
10 REM SPACE LANDING SIMULATION I
20 PRINT RND(TIME)
30 REM ********************************
40 REM SET STARTING VALUES
50 FUEL=200+RND(1)*40
60 VELOCITY=RND(1)*20-6
70 HEIGHT=500-RND(1)*10
80 MODE 6:VDU 19,1,2;0;
90 PRINT " FUEL";TAB(12);" VELOCITY";TAB(24);" HEIGHT"
100 REM ********************************
110 REM MAJOR CYCLE
120  PROCpicture
130 IF FUEL<=0 THEN FUEL=0:THRUST=0:GOTO 170
140 A$=INKEY$(0)
150 IF A$<"0" OR A$>"9" THEN 140
160 THRUST=VAL(A$)+.1
170 FUEL=FUEL-THRUST
180 FLAG=THRUST-2
190 THRUST=0
200 HEIGHT=HEIGHT+VELOCITY+FLAG/4
210 VELOCITY=VELOCITY+FLAG
220 IF HEIGHT<=10 THEN 240
230 IF HEIGHT>10 THEN 120
240 IF VELOCITY>-9 AND VELOCITY<5 THEN 290
250  PROCrule
260 PRINT "YOU HAVE CRASHED INTO THE SURFACE..."
270 IF HEIGHT>0 THEN HEIGHT=-HEIGHT
280 GOTO 320
290 PRINT "YOU HAVE LANDED SAFELY!"
300 PRINT "YOUR SKILL RATING IS"INT(-1000*FUEL/(VELOCITY-HEI
GHT))
310 HEIGHT=0
320  PROCrule
330 PRINT "FINAL INSTRUMENT READINGS WERE:"
340 PRINT " FUEL";TAB(12);" VELOCITY";TAB(24);" HEIGHT"
```

```
350  PROCpicture
360  PROCrule
370 IF HEIGHT>=0 THEN END
380 IF HEIGHT<0 THEN PRINT "NEW CRATER ON MOON"INT(ABS(100*(
HEIGHT+.2)/3))/100"METERS DEEP!"
390 PRINT "YOUR SKILL RATING IS "INT(100*FUEL/(VELOCITY-HEIG
HT))
400 END
410  DEFPROCrule
420 PRINT "----------------------------------------"
430  ENDPROC
440  DEFPROCpicture
450 PRINT ;INT(100*FUEL)/100;
460 PRINT TAB(12);-INT(100*VELOCITY)/100;
470 IF HEIGHT>=0 THEN PRINT TAB(24);INT(100*HEIGHT)/100
480 IF HEIGHT<0 THEN PRINT
490  ENDPROC
```

Monte Carlo

```
10 REM MONTE CARLO DEMONSTRATION
20  PROCinitial
30 REM ********************************
40 REM MAJOR CYCLE
50 REPEAT
60  PROCpicture
70 IF P=EP AND Q=EQ THEN PRINT:PRINT "DEMONSTRATION OVER":END
80  PROCmonte
90 UNTIL FALSE
100 REM ********************************
110  DEFPROCpicture
120 A$(P,Q)="O"
130 M=M+1
140 CLS:PRINT:PRINT
150 PRINT "MOVE"M
160 FOR X=1 TO 10
170 FOR Y=1 TO 10
180 PRINT A$(X,Y);" ";
190 NEXT Y
200 PRINT
210 NEXT X
220  ENDPROC
230 REM ********************************
240  DEFPROCmonte
250 A$(P,Q)="."
260 G=0
270 T=RND(4)
280 ON T GOSUB 320,330,340,350
290 IF G=0 THEN 270
300 IF G=1 AND RND(1)>.5 THEN 270
310 RETURN
320 IF P>1 THEN P=P-1:G=G+1:RETURN
330 IF P<10 THEN P=P+1:G=G+1:RETURN
340 IF Q>1 THEN Q=Q-1:G=G+1:RETURN
350 IF Q<10 THEN Q=Q+1:G=G+1:RETURN
360  ENDPROC
370 REM ********************************
380  DEFPROCinitial
390 PRINT RND(TIME)
```

```
400 MODE 4:VDU 19,0,7;0;19,1,1;0;
410 DIM A$[10,10]
420 FOR X=1 TO 10
430 FOR Y=1 TO 10
440 A$[X,Y]="."
450 NEXT Y
460 NEXT X
470 PRINT:PRINT
480 PRINT "ENTER FIRST START CO-ORDINATE [LESS     THAN 10]"
490 INPUT P
500 IF P<1 OR P>10 THEN 490
510 PRINT "ENTER SECOND START CO-ORDINATE [LESS    THAN 10]"
520 INPUT Q
530 IF Q<1 OR Q>10 THEN 520
540 PRINT:PRINT
550 PRINT "ENTER FIRST END CO-ORDINATE [LESS      THAN 10]"
560 INPUT EP
570 IF EP=P OR EP<1 OR EP>10 THEN 560
580 PRINT "ENTER SECOND END CO-ORDINATE [LESS     THAN 10]"
590 INPUT EQ
600 IF EQ=Q OR EQ<1 OR EQ>10 THEN 590
610 A$[P,Q]="O"
620 A$[EP,EQ]="X"
630   ENDPROC
```

Simultaneous Equations

```
10 REM SIMULTANEOUS EQUATIONS
20 PRINT RND(TIME)
30 MODE 6:VDU 19,1,3;0;
40 HS=0
50 FD=RND(1)
60 PRINT:PRINT "DECAY FACTOR IS"FD
70  PROCdelay
80 CLS
90 PRINT:PRINT
100 PRINT "ENTER NUMBER OF CELL X TO                    START (LE
SS THAN 40)"
110 INPUT CP:IF CP<1 OR CP>39 THEN 110
120 PRINT:PRINT
130 PRINT "WE HAVE"CP"X CELLS"
140 PRINT:PRINT
150 PRINT "ENTER NUMBER OF CELL Y TO                    START (LE
SS THAN 40)"
160 INPUT EP:IF EP<1 OR EP>39 THEN 160
170 CLS:PRINT:PRINT "PLEASE STAND BY..."
180  PROCdelay:CLS
190 DA=1
200 IF CP>EP/FD THEN CP=EP/FD
210 PRINT "_____________________"
220 PRINT "TIME ELAPSED:"DA
230 PRINT INT(CP)"CELL X"
240 PRINT INT(EP)"CELL Y"
250 REM ******************************
260 REM MAJOR CYCLE
270  PROCdelay
280 DA=DA+1
290 PRINT "_____________________"
300 PRINT "TIME ELAPSED:"DA
310 IF CP>EP/FD THEN CP=EP/FD
320 REM EQUATIONS FOLLOW; MODIFY PARTS OF THEM TO SEE WHAT
HAPPENS
330 CP=CP+((8*CP-CP*EP/3)*FD)
340 EP=EP+((4*EP-EP*CP)*.01)
350 PRINT INT(CP)"CELL X"
```

```
360 PRINT INT[EP]"CELL Y"
370 IF EP<2 OR CP<2 THEN 410
380  PROCdelay
390 GOTO 280
400 REM ******************************
410 IF DA>HS THEN HS=DA
420 PRINT:PRINT
430 PRINT "YOUR CELL CLASH SIMULATION SURVIVED"
440 PRINT "FOR"DA"TIME PERIODS."
450 PRINT "_____________________________________"
460 PRINT "THE BEST SURVIVAL TIME SO FAR IS"HS
470  PROCdelay
480 PRINT "_____________________________________"
490 PRINT "DO YOU WANT A NEW RUN [Y OR N]?"
500 A$=GET$
510 IF A$<>"Y" AND A$<>"N" THEN 500
520 IF A$="Y" THEN CLS:GOTO 60
530 PRINT "OK":PRINT:PRINT:END
540 REM ******************************
550  DEFPROCdelay
560 FOR J=1 TO 2000:NEXT J
570  ENDPROC
```

Life

```
10 REM CONWAY'S LIFE SIMULATION
20 REM RANDOM INITIAL COLONY
30  PROCinitial
40 REM ****************************
50 REM MAJOR CYCLE
60 REPEAT
70 GENERATION=GENERATION+1
80  PROCpicture
90  PROCevolve
100 UNTIL FALSE
110 REM ****************************
120  DEFPROCevolve
130 FOR X=2 TO 12
140 FOR Y=2 TO 12
150 C=0
160 IF A$(X-1,Y-1)="X" THEN C=C+1
170 IF A$(X-1,Y)="X" THEN C=C+1
180 IF A$(X-1,Y+1)="X" THEN C=C+1
190 IF A$(X,Y-1)="X" THEN C=C+1
200 IF A$(X,Y+1)="X" THEN C=C+1
210 IF A$(X+1,Y-1)="X" THEN C=C+1
220 IF A$(X+1,Y)="X" THEN C=C+1
230 IF A$(X+1,Y+1)="X" THEN C=C+1
240 IF A$(X,Y)="X" AND C<>2 AND C<>3 THEN B$(X,Y)=" "
250 IF A$(X,Y)=" " AND C=3 THEN B$(X,Y)="X"
260 NEXT Y
270 NEXT X
280  ENDPROC
290 REM ****************************
300  DEFPROCpicture
310 CLS
320 PRINT
330 PRINT TAB(4);"GENERATION"GENERATION
340 PRINT
350 FOR X=2 TO 12
360 FOR Y=2 TO 12
370 A$(X,Y)=B$(X,Y)
380 PRINT A$(X,Y);
```

```
390 NEXT Y
400 FOR Y=12 TO 2 STEP-1
410 PRINT A$(X,Y);
420 NEXT Y
430 PRINT
440 NEXT X
450  ENDPROC
460 REM ****************************
470  DEFPROCinitial
480 PRINT RND(TIME)
490 VDU 19,0,1;0;19,1;3;0;
500 DIM A$(13,13),B$(13,13)
510 PRINT:PRINT "   PLEASE STAND BY..."
520 FOR X=1 TO 13
530 PRINT 14-X;
540 FOR Y=1 TO 13
550 IF X=1 OR Y=1 OR X=13 OR Y=13 THEN 570
560 IF RND(1)>.5 THEN A$(X,Y)="X":GOTO 580
570 A$(X,Y)=" "
580 B$(X,Y)=A$(X,Y)
590 NEXT Y
600 NEXT X
610 GOTO 650
620 READ D:IF D=99 THEN 650
630 READ E:A$(D,E)="X":B$(D,E)="X"
640 GOTO 620
650 GENERATION=0
660  ENDPROC
670 DATA 5,5,5,9,6,6,6,8
680 DATA 7,7
690 DATA 8,6,8,8,9,5,9,9
700 DATA 99
```

Robot Logo

```
100 REM ROBOT LOGO
105 REM BBC VERSION
107 MODE 7
110 PROCinitialise
120 GOTO 490
130 REM ************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ************************
340 REM INT UX,AX
350 UX=INT(UX+.5):AX=INT(AX+.5)
360 RETURN
370 REM ************************
380 REM       PRINT OUT
390 CLS
400 PRINT "STEP"PSN" > ";A$(PSN):PRINT
410 FOR J=1 TO DEPTH
420 FOR K=1 TO BREADTH
430 PRINT Z$(J,K);
440 NEXT K
450 PRINT
460 NEXT J
```

```
470 RETURN
480 REM ************************
490 REM READ PROGRAM
500 CNT=CNT+1
510 READ A$(CNT)
520 IF A$(CNT)="*" THEN 550
530 IF CNT<20 THEN 500
540 REM ************************
550 REM EXECUTE PROGRAM
560 PSN=0:REM PROGRAM STEP NUMBER
570 PSN=PSN+1
580 IF PSN=21 THEN 580:REM END
590 FLAG=0
600 M$=A$(PSN)
610 IF M$="*" THEN 610:REM END
620 N$=LEFT$(M$,2)
630 IF N$="ST" THEN 560:REM START AGAIN
640 IF N$="PR" THEN GOSUB 380:REM PRINTOUT
650 IF N$="FO" THEN FLAG=1
660 IF N$="BA" THEN FLAG=2
670 IF N$="TU" THEN FLAG=3
680 IF N$="HO" THEN FLAG=4
690 IF N$="CL" THEN FLAG=5
700 IF N$="GO" THEN FLAG=6
710 IF N$="RA" THEN FLAG=7
720 IF N$="RE" THEN FLAG=8
730 IF N$="EN" THEN FLAG=9
740 IF N$="FA" THEN FLAG=10
750 IF FLAG<1 THEN 570
760 ON FLAG GOSUB 780,940,1000,1160,1220,1260,1400,1460,1530
,1580
765 GOTO 570
770 REM ************************
780 REM          FORWARD
790 M$=MID$(M$,4)
800 IF ASC(M$)=87 THEN M$=MID$(M$,6)
810 F$="F"
820 NUM=VAL(M$)
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN 880
850 IF AX<1 OR UX>BREADTH THEN 880
```

```
860 Z$(UX,AX)=T$
880 IF F$="F" THEN UX=UX+UP:AX=AX+AC*2
890 IF F$="B" THEN UX=UX-UP:AX=AX-AC*2
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ************************
940 REM         BACK
 950 M$=MID$(M$,4)
 960 IF ASC(M$)=75 THEN M$=MID$(M$,3)
 970 F$="B"
 980 GOTO 820
 990 REM ************************
1000 REM         TURN
1010 M$=MID$(M$,4)
1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
1030 NUM=VAL(M$)
1040 Y=INT((NUM+17.5)/45)
1050 IF Y=0 OR Y=8 THEN RETURN
1060 FOR J=1 TO Y
1070 IF UP=-1 AND AC=0 THEN AC=1:GOTO 1130
1080 IF UP=0 AND AC=1 THEN UP=1:GOTO 1130
1090 IF UP=1 AND AC=0 THEN AC=-1:GOTO 1130
1100 IF UP=0 AND AC=-1 THEN UP=-1:GOTO 1130
1110 IF (UP=-1 AND AC=-1) OR (UP=1 AND AC=1) THEN AC=0:GOTO 1
130
1120 IF (UP=-1 AND AC=1) OR (UP=1 AND AC=-1) THEN UP=0
1130 NEXT J
1140 RETURN
1150 REM ************************
1160 REM         HOME
1170 AX=INT((BREADTH+.5)/2)
1180 UX=INT((DEPTH+.5)/2)
1190 UP=-1:AC=0:REM FACES UP
1200 RETURN
1210 REM ************************
1220 REM         CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ************************
1260 REM         GO X,Y
```

```
1270 P=0
1280 P=P+1
1290 IF MID$(M$,P,1)="," THEN 1320
1300 IF P<LEN(M$) THEN 1280
1310 RETURN:REM ERROR
1320 UX=VAL(MID$(M$,4,P-1))
1330 AX=VAL(RIGHT$(M$,LEN(M$)-P))
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
1360 IF AX<1 OR AX>BREADTH THEN 1380
1370 Z$(UX,AX)=R$
1380 RETURN
1390 REM ************************
1400 REM          RANDOM
1410 AX=INT(RND(1)*BREADTH)
1420 UX=INT(RND(1)*DEPTH)
1430 Z$(UX,AX)=R$
1440 RETURN
1450 REM ************************
1460 REM          REPEAT
1470 M$=MID$(M$,4)
1480 IF ASC(M$)=69 THEN M$=MID$(M$,5)
1490 RECNT=VAL(M$)
1500 MARKER=PSN
1510 RETURN
1520 REM ************************
1530 REM        END REPEAT
1540 RECNT=RECNT-1
1550 IF RECNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ************************
1580 REM          FACE
1590 M$=MID$(M$,4)
1600 IF ASC(M$)=69 THEN M$=MID$(M$,3)
1610 NUM=VAL(M$)
1620 Y=INT((NUM+17.5)/45)*45
1630 IF Y=0 OR Y=360 THEN UP=-1:AC=0
1640 IF Y=45 THEN UP=-1:AC=1
1650 IF Y=90 THEN UP=0:AC=1
1660 IF Y=135 THEN UP=1:AC=1
1670 IF Y=180 THEN UP=1:AC=0
```

```
1680 IF Y=225 THEN UP=1:AC=-1
1690 IF Y=270 THEN UP=0:AC=-1
1700 IF Y=315 THEN UP=-1:AC=-1
1710 RETURN
1720 REM ************************
1730 DEFPROCinitialise
1740 PRINT RND(TIME)
1750 CLS
1760 REM ADJUST NEXT TWO LINES FOR DIFFERENT MODES
1770 BREADTH=40:REM CHARACTERS ACROSS
1780 DEPTH=25:REM CHARACTERS DOWN
1790 BREADTH=BREADTH-1
1800 DEPTH=DEPTH-3
1810 UP=-1:AC=0:REM STARTS FACING UP
1820 DIM A$(20):REM FOR ROBOT PROGRAM
1830 DIM Z$(DEPTH,BREADTH):REM DISPLAY
1840 T$="X":R$="X":REM SYMBOL FOR ROBOT'S TRAIL
1850 AX=0:UX=0
1860 REM FILL ARRAY WITH SPACES
1870 FOR J=1 TO DEPTH
1880 FOR K=1 TO BREADTH
1890 Z$(J,K)=" "
1900 NEXT K
1910 NEXT J
1920 ENDPROC
```

Point Duty Robot

```
100 REM POINT-DUTY ROBOT
105 REM BBC VERSION
107 MODE 0
110 PROCinitialise
120 GOTO 490
130 REM ************************
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ************************
340 REM INT UX,AX
350 UX=INT(UX+.5):AX=INT(AX+.5)
360 RETURN
480 REM ************************
490 REM READ PROGRAM
500 CNT=CNT+1
510 READ A$(CNT)
520 IF A$(CNT)="*" THEN 550
530 IF CNT<20 THEN 500
540 REM ************************
550 REM EXECUTE PROGRAM
560 PSN=0:REM PROGRAM STEP NUMBER
570 PSN=PSN+1
```

```
580 IF PSN=21 THEN 580:REM END
590 FLAG=0
600 M$=A$[PSN]
610 IF M$="*" THEN 610:REM END
620 N$=LEFT$[M$,2]
630 IF N$="ST" THEN 560:REM START AGAIN
650 IF N$="FO" THEN FLAG=1
660 IF N$="BA" THEN FLAG=2
670 IF N$="TU" THEN FLAG=3
680 IF N$="HO" THEN FLAG=4
700 IF N$="GO" THEN FLAG=5
710 IF N$="RA" THEN FLAG=6
720 IF N$="RE" THEN FLAG=7
730 IF N$="EN" THEN FLAG=8
740 IF N$="FA" THEN FLAG=9
750 IF FLAG<1 THEN 570
760 ON FLAG GOSUB 780,940,1000,1160,1260,1400,1460,1530,1580
765 GOTO 570
770 REM ************************
780 REM         FORWARD
790 M$=MID$[M$,4]
800 IF ASC[M$]=87 THEN M$=MID$[M$,6]
810 F$="F"
820 NUM=VAL[M$]
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN 880
850 IF AX<1 OR UX>BREADTH THEN 880
860 PLOT 69,AX,UX
880 IF F$="F" THEN UX=UX+UP:AX=AX+AC*2
890 IF F$="B" THEN UX=UX-UP:AX=AX-AC*2
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ************************
940 REM         BACK
950 M$=MID$[M$,4]
960 IF ASC[M$]=75 THEN M$=MID$[M$,3]
970 F$="B"
980 GOTO 820
990 REM ************************
1000 REM         TURN
```

```
 1010 M$=MID$(M$,4)
 1020 IF ASC(M$)=78 THEN M$=MID$(M$,3)
 1030 NUM=VAL(M$)
 1040 Y=INT((NUM+11.25)/22.5)
 1050 IF Y=0 OR Y=16 THEN RETURN
 1060 FOR J=1 TO Y
 1065 IF (UP=-2 AND AC=0) OR (UP=2 AND AC=2) THEN AC=1:GOTO 11
30
 1070 IF UP=-2 AND AC=1 THEN AC=2:GOTO 1130
 1075 IF (UP=-2 AND AC=2) OR (UP=0 AND AC=-2) THEN UP=-1:GOTO
1130
 1080 IF (UP=-1 AND AC=2) OR (UP=1 AND AC=-2) THEN UP=0:GOTO 1
130
 1085 IF (UP=0 AND AC=2) OR (UP=2 AND AC=-2) THEN UP=1:GOTO 11
30
 1090 IF UP=1 AND AC=2 THEN UP=2:GOTO 1130
 1095 IF UP=2 AND AC=1 THEN AC=0:GOTO 1130
 1100 IF UP=2 AND AC=0 THEN AC=-1:GOTO 1130
 1105 IF UP=2 AND AC=-1 THEN AC=-2:GOTO 1130
 1110 IF UP=-1 AND AC=-2 THEN UP=-2:GOTO 1130
 1115 IF UP=-2 AND AC=-2 THEN AC=-1:GOTO 1130
 1120 IF UP=-2 AND AC=-1 THEN AC=0
 1130 NEXT J
 1140 RETURN
 1150 REM ************************
 1160 REM         HOME
 1170 AX=INT((BREADTH+.5)/2)
 1180 UX=INT((DEPTH+.5)/2)
 1190 UP=-2:AC=0:REM FACES UP
 1200 RETURN
 1210 REM ************************
 1220 REM         CLEAN
 1230 GOSUB 1870
 1240 RETURN
 1250 REM ************************
 1260 REM         GO X,Y
 1270 P=0
 1280 P=P+1
 1290 IF MID$(M$,P,1)="," THEN 1320
 1300 IF P<LEN(M$) THEN 1280
 1310 RETURN:REM ERROR
```

```
1320 UX=VAL[MID$[M$,4,P-1]]
1330 AX=VAL[RIGHT$[M$,LEN[M$]-P]]
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN 1380
1360 IF AX<1 OR AX>BREADTH THEN 1380
1370 PLOT 69,AX,UX
1380 RETURN
1390 REM ************************
1400 REM          RANDOM
1410 AX=INT[RND[1]*BREADTH]
1420 UX=INT[RND[1]*DEPTH]
1430 PLOT 69,AX,UX
1440 RETURN
1450 REM ************************
1460 REM          REPEAT
1470 M$=MID$[M$,4]
1480 IF ASC[M$]=69 THEN M$=MID$[M$,5]
1490 RECNT=VAL[M$]
1500 MARKER=PSN
1510 RETURN
1520 REM ************************
1530 REM          END REPEAT
1540 RECNT=RECNT-1
1550 IF RECNT>0 THEN PSN=MARKER
1560 RETURN
1570 REM ************************
1580 REM          FACE
1590 M$=MID$[M$,4]
1600 IF ASC[M$]=69 THEN M$=MID$[M$,3]
1610 NUM=VAL[M$]
1620 Y=INT[[NUM+11.25]/22.5]*22.5
1630 IF Y=0 OR Y=360 THEN UP=-2:AC=0
1635 IF Y=22.5 THEN UP=-2:AC=1
1640 IF Y=45 THEN UP=-2:AC=2
1645 IF Y=67.5 THEN UP=-1:AC=2
1650 IF Y=90 THEN UP=0:AC=2
1655 IF Y=112.5 THEN UP=1:AC=2
1660 IF Y=135 THEN UP=2:AC=2
1665 IF Y=157.5 THEN UP=2:AC=1
1670 IF Y=180 THEN UP=2:AC=0
1675 IF Y=202.5 THEN UP=2:AC=-1
```

```
1680 IF Y=225 THEN UP=2:AC=-2
1685 IF Y=247.5 THEN UP=1:AC=-2
1690 IF Y=270 THEN UP=0:AC=-2
1695 IF Y=292.5 THEN UP=-1:AC=-2
1700 IF Y=315 THEN UP=-2:AC=-2
1705 IF Y=337.5 THEN UP=-2:AC=-1
1710 RETURN
1720 REM ************************
1730 DEFPROCinitialise
1740 PRINT RND(TIME)
1750 CLS
1760 REM ADJUST NEXT TWO LINES FOR DIFFERENT MODES
1770 BREADTH=1280:REM CHARACTERS ACROSS
1780 DEPTH=1024:REM CHARACTERS DOWN
1790 BREADTH=BREADTH-1
1800 DEPTH=DEPTH-3
1810 UP=-2:AC=0:REM STARTS FACING UP
1820 DIM A$(20):REM FOR ROBOT PROGRAM
1830 UX=0:AX=UX
1840 ENDPROC
```

Connect Four

```
10 REM CONNECT FOUR
20 REM BBC version
30 REM A. W. PEARSON
40 MODE 6:VDU 19,0,4;0;
50 PRINT
60 PRINT
70 PRINT "CONNECT FOUR"
80 PRINT
90 PRINT "ENTER YOUR MOVE AS A NUMBER BETWEEN"
100 PRINT "1 AND 8, ENTER 0 FOR A NEW GAME..."
110 FOR F=1 TO 1000:NEXT F
120 DIM A$(10,10),B(10,2)
130 FLAG=0
140 C$="M":H$="o"
150 FOR F=1 TO 8
160 B(F,1)=6
170 NEXT F
180 FOR F=1 TO 6
190 FOR G=1 TO 8
200 A$(F,G)="."
210 NEXT G
220 NEXT F
230 REM ***************************
240 REM ACCEPT HUMAN MOVE
250  PROCboard
260 PRINT:PRINT "YOUR MOVE..."
270 INPUT A
280 IF A=0 THEN RUN
290 IF A<1 OR A>8 THEN 270
300 L=0
310 IF A$(L+1,A)<>"." OR L=6 THEN 340
320 L=L+1
330 GOTO 310
340 IF L=0 THEN 270
350 A$(L,A)=H$
360 B(A,1)=B(A,1)-1
370  PROCboard
380  PROCmoves
390  PROCboard
```

```
400 GOTO 260
410 REM ****************************
420  DEFPROCboard
430 CLS
440 FOR F=1 TO 6
450 FOR G=1 TO 8
460 PRINT A$(F,G);
470 NEXT G
480 PRINT
490 NEXT F
500 PRINT "12345678"
510 PRINT
520 IF FLAG=1 THEN PRINT "I HAVE WON":END
530  ENDPROC
540 REM ****************************
550  DEFPROCmoves
560 PRINT "MY MOVE..."
570 MV=0
580 FOR F=1 TO 8
590 B(F,2)=0
600 NEXT F
610 FOR F=1 TO 8
620 FOR X=-1 TO 1
630 FOR Y=-1 TO 1
640 IF B(F,1)=0 THEN 680
650 IF A$(B(F,1)+X,F+Y)="" OR A$(B(F,1)+X,F+Y)="." THEN 680
660 IF A$(B(F,1)+X,F+Y)=H$ THEN GOSUB 810
670 IF A$(B(F,1)+X,F+Y)=C$ THEN GOSUB 910
680 NEXT Y
690 NEXT X
700 NEXT F
710 P=0
720 FOR F=1 TO 8
730 IF B(F,2)>P THEN P=B(F,2):N=F
740 NEXT F
750 A$(B(N,1),N)=C$
760 B(N,1)=B(N,1)-1
770 N=0
780 P=0
790  ENDPROC
800 REM ****************************
```

```
810 MV=2
820 M1=MV
830 IF A$[B[F,1]+[X*2],F+[Y*2]]=H$ THEN MV=MV+10
840 IF A$[B[F,1]-X,F-Y]=H$ THEN MV=MV+20
850 IF MV<>M1+10 THEN 870
860 IF A$[B[F,1]+[X*3],F+[Y*3]]=H$ THEN MV=MV+1000
870 B[F,2]=B[F,2]+MV
880 M1=0
890 RETURN
900 REM ***************************
910 MV=2
920 M1=MV
930 IF A$[B[F,1]+[X*2],F+[Y*2]]=C$ THEN MV=MV+9
940 IF A$[B[F,1]-X,F-Y]=C$ THEN MV=MV+20
950 IF MV<>M1+9 THEN 970
960 IF A$[B[F,1]+[X*3],F+[Y*3]]=C$ THEN MV=MV+2000:FLAG=1
970 B[F,2]=B[F,2]+MV
980 RETURN
```

Washington D.C.

```
10 REM WASHINGTON D.C. - BBC version
20 GOSUB 1160:REM INITIALIZE
30 REM ******************************
40 REM MAJOR CYCLE
50 P=INT(P+(P*273/ML))
60 GOSUB 160:REM PRINTOUT
70 GOSUB 510:REM CALCULATE
80 REM NOW CHECK END GAME
90 GOSUB 710:REM STANDARD OF LIVING
100 GOSUB 780:REM INFLATION RATE
110 GOSUB 840:REM UNEMPLOYMENT
120 IF GAME=1 THEN CLS:GOTO 890
130 GOTO 50
140 REM ******************************
150 REM PRINTOUT
160 CLS
170 PRINT "PRESIDENT ";A$;":"
180 PRINT "YOUR ADMINISTRATION HAS BEEN IN                    PO
WER FOR"Y+Z/4"YEARS"
190 PRINT "-----------------------------------------"
200 PRINT "------STATE OF THE NATION----------"
210 PRINT "-----------------------------------------"
220 PRINT "POPULATION"P
230 PRINT "NO. UNEMPLOYED"INT(U)"  "INT(100*U/P)"%"
240 PRINT "CURRENT WAGES $"WO" INFLATION"INT(IP)"%"
250 PRINT "GOVT. EXPENDITURE LAST QTR. $M"GE
260 PRINT "UNEMPLOYMENT COST $M"INT(10*GU)/10
270 PRINT "INCOME FROM TAXES $M"INT(GI*10)/10
280 PRINT "BUDGET SURPLUS(+)/DEFICIT(-) $M"INT(BD*10)/10
290 PRINT "GROSS DOMESTIC PRODUCT $M"INT(GDP*10)/10
300 IF Y+Z/4>.5 THEN PRINT "CHANGE IN LIVING STANDARD"INT((2
*((RGDP/AGDP)*100)-100)/3)"%"
310 PRINT "-----------------------------------------"
320 PRINT "PUBLIC INVESTMENT"Y"Q"Z"$M"INT(IV*10)/10
330 PRINT "-----------------------------------------"
340 PRINT "OK, PRESIDENT ";A$;"..."
350 INPUT "ENTER GOVERNMENT SPENDING $M";GE
360 INPUT "ENTER COST OF WAGES $M";WN
370 PRINT "IS YOUR ADMINISTRATION IN FAVOR OF"
```

```
380 PRINT "IMMIGRATION THIS QUARTER (Y/N)?"
390 X$=INKEY$
400 IF X$<>"Y" AND X$<>"N" THEN 390
410 PRINT TAB(20);"OK...";X$
420 FOR H=1 TO 1000:NEXT H
430 IF X$<>"Y" THEN RETURN
440 PRINT "HOW MANY IMMIGRANTS WILL YOU ALLOW                    I
NTO THE US?"
450 INPUT M
460 IF M<0 THEN 450
470 P=P+M
480 RETURN
490 REM ******************************
500 REM CALCULATIONS
510 CN=CN+(CN*IP/100)
520 U=P*(GE+IV)/(CN*10)+P*(IP/1000)
530 GU=U*WN/ML:REM UNEMPLOYMENT COST
540 GI=(((P-U)*WN*.4)/ML):REM INCOME
      FROM TAXES
550 BD=BD+GI-GU-GE:REM BUDGET DEFICIT
560 AGDP=AGDP*(1+(IP/100))
570 GDP=GE+IV+((P-U)*WN/ML)
580 RGDP=GDP*440/AGDP
590 IP=((GE+IV)/CN*.1+(WN/WO)/100)*100
600 IV=(CN*67)/(IP*IP)
610 WO=WN
620 Z=Z+1:IF Z>4 THEN Z=1:Y=Y+1
630 RETURN
640 REM ******************************
650 REM CHECK BUDGET DEFICIT
660 IF BD>-1000 THEN RETURN
670 GAME=1
680 FLAG=1
690 RETURN
700 REM ******************************
710 REM CHECK STANDARD OF LIVING
720 IF Y<.75 THEN RETURN
730 IF INT((2*((RGDP/AGDP)*100)-100)/3)>-15 THEN RETURN
740 GAME=1
750 FLAG=2
760 RETURN
```

```
770 REM ********************************
780 REM CHECK INFLATION RATE
790 IF IP<15 THEN RETURN
800 GAME=1
810 FLAG=3
820 RETURN
830 REM ********************************
840 REM CHECK UNEMPLOYMENT
850 IF INT(U*100/P)<15 THEN RETURN
860 GAME=1
870 FLAG=4
880 RETURN
890 REM ********************************
900 REM END OF THE GAME
910 PRINT "PRESIDENT ";A$;", YOUR"
920 PRINT "ADMINISTRATION'S POOR ECONOMIC"
930 PRINT "PERFORMANCE HAS LED TO AN UNACCEPTABLE"
940 IF FLAG=1 THEN PRINT "BUDGET DEFICIT"
950 IF FLAG=2 THEN PRINT "DROP IN THE STANDARD OF LIVING"
960 IF FLAG=3 THEN PRINT "RISE IN THE INFLATION RATE"
970 IF FLAG=4 THEN PRINT "RISE IN UNEMPLOYMENT"
980 PRINT "          AMONG OTHER THINGS..."
990 PRINT "_____________________________________"
1000 PRINT "THE LACK OF CONFIDENCE IN YOUR"
1010 PRINT "ADMINISTRATION IS SO BAD THERE ARE"
1020 PRINT "CALLS FOR YOU TO RESIGN...YOU STEP"
1030 PRINT "ASIDE TO ALLOW THE VICE-PRESIDENT TO"
1040 PRINT "       OCCUPY THE OVAL OFFICE"
1050 FOR H=1 TO 1000:NEXT H
1060 PRINT "_____________________________________"
1070 PRINT "YOU WERE PRESIDENT FOR"Y+(Z*.25)"YEARS"
1080 PRINT "DURING YOUR TERM OF OFFICE, THE"
1090 PRINT "POPULATION ROSE BY"P-3*ML
1100 PRINT "THE UNEMPLOYMENT RATE BECAME"INT(U*1000/P)/10"%"
1110 PRINT "AND THE INFLATION RATE BECAME"INT(10*IP)/10"%"
1120 PRINT "STANDARD OF LIVING CHANGED BY"INT((2*((RGDP/AGDP
)*100)-100)/3)"%"
1130 PRINT "AND THE BUDGET SURPLUS/DEFICIT          WAS $M"I
NT(10*BD)/10
1140 END
1150 REM ********************************
```

```
1160 REM INITIALIZATION
1170 PRINT RND(TIME)
1180 MODE 6:VDU 19,0,3;0;19,1,0;0;
1190 ML=1000*1000
1200 P=3*ML:REM POPULATION
1210 U=P/10:REM UNEMPLOYMENT
1220 IV=236:REM INVESTMENT
1230 GE=118:REM GOVERNMENT EXPENDITURE
1240 GU=0:REM COST OF UNEMPLOYMENT
1250 GI=0:REM INCOME FROM TAXES
1260 WN=100:REM NEW WAGES
1270 WO=100:REM OLD WAGES
1280 IP=5:REM INFLATION PERCENT
1290 GDP=440:REM GROSS DOMESTIC PRODUCT
1300 AGDP=440:REM BASE YEAR GDP
1310 RGDP=440:REM REAL GDP
1320 CN=354:REM ECONOMIC CONSTANT (USED THROUGHOUT)
1330 Z=1:GAME=0:FLAG=0
1340 Y=0:REM YEAR
1350 PRINT "ENTER YOUR LAST NAME"
1360 INPUT A$
1370 RETURN
```

Stock Market

```
10 REM STOCK MARKET
20 REM BBC version
30 PRINT RND(TIME)
40 MODE 6:VDU 19,0,3;0;19,1,1;0;
50 DIM S(5),N(5),P(5),D(5)
60 S(1)=1.49:S(2)=1.99:S(3)=2.49:S(4)=2.99:S(5)=3.49
70 N(1)=2000:N(2)=1500:N(3)=1200:N(4)=1000:N(5)=800
80 BB=265:TV=15000:QQ=15000:DAY=1
90 PRINT:PRINT "ENTER YOUR GOAL FOR THIS SIMULATION,"
100 PRINT TAB(8);"$16,000 TO $100,000"
110 INPUT GOAL
120 IF GOAL<16000 THEN PRINT "TOO LOW!":GOTO 90
130 IF GOAL>100*1000 THEN PRINT "TOO HIGH!":GOTO 90
140 REM ***************************
150 REM MAJOR LOOP
160 FOR C=1 TO 5
170 REM ADJUST THE 55 IN NEXT LINE TO MODIFY GAME;  80 VERY
HARD, 30 VERY EASY
180 D(C)=RND(55)
190 P(C)=RND(100-D(C))
200 NEXT C
210  PROCpicture
220 GOTO 470
230 REM ***************************
240  DEFPROCpicture
250 CLS
260 PRINT "_____________________________________________"
270 PRINT "DAY"DAY"        YOUR GOAL IS  $"GOAL
280 PRINT "_____________________________________________"
290 PRINT "COMPANY NUMBER:"
300 PRINT TAB(2);1;TAB(9);2;TAB(16);3;TAB(25);4;TAB(32);5
310 PRINT "CHANCE OF INCREASE (%):"
320 PRINT TAB(2);P(1);TAB(9);P(2);TAB(16);P(3);TAB(25);P(4);
TAB(32);P(5)
330 PRINT "CHANCE OF DECREASE (%):"
340 PRINT TAB(2);D(1);TAB(9);D(2);TAB(16);D(3);TAB(25);D(4);
TAB(32);D(5)
350 PRINT "CURRENT VALUE PER SHARE:"
360 PRINT "$";INT(S(1)*100)/100;TAB(8);"$";INT(S(2)*100)/100
;
```

```
370 PRINT TAB(15);"$";INT(S(3)*100)/100;TAB(23);"$";INT(S(4)
*100)/100;
380 PRINT TAB(30);"$";INT(S(5)*100)/100
390 PRINT "NO. OF SHARES HELD:"
400 PRINT TAB(2);N(1);TAB(9);N(2);TAB(16);N(3);TAB(24);N(4);
TAB(31);N(5)
410 PRINT "BANK $"INT(BB)" TOTAL WORTH $"TV
420 PRINT "_____"
430 IF TV>GOAL THEN PRINT "YOU'VE HIT YOUR FINANCIAL GOAL!":
END
440  ENDPROC
450 REM **************************
460 REM          ** SELL **
470 PRINT "DO YOU WANT TO SELL ANY SHARES (Y/N)?"
480 A$=GET$
490 IF A$<>"Y" AND A$<>"N" THEN 480
500 IF A$="N" THEN 700
510  PROCpicture
520 PRINT "WHICH ONES TO SELL?";
530 A$=INKEY$
540 IF A$<"1" OR A$>"5" THEN 530
550 C=VAL(A$)
560 PRINT "     OK"C
570 PRINT "HOW MANY OF"C"TO SELL";
580 INPUT N
590 IF N>N(C) THEN PRINT "YOU DON'T HAVE THAT MANY!":GOTO 58
0
600 REM **************************
610 REM ADJUST FIGURES AFTER SALE
620 BB=BB+S(C)*N:REM ADD VALUE TO BANK
630 N(C)=N(C)-N:REM SUBTRACT NO. SOLD
640 TV=0:REM SET TOTAL WORTH TO ZERO
650 REM NOW DETERMINE CURRENT WORTH
660 FOR C=1 TO 5
670 TV=TV+N(C)*S(C)
680 NEXT C
690 TV=TV+BB:REM ADD IN BANK BALANCE
700  PROCpicture
710 REM **************************
720 REM          ** BUY **
730 PRINT "DO YOU WANT TO BUY ANY SHARES (Y/N)?"
```

```
740 A$=GET$
750 IF A$<>"Y" AND A$<>"N" THEN 740
760 IF A$="N" THEN 900
770  PROCpicture
780 PRINT "WHICH COMPANY TO BUY?";
790 A$=GET$
800 IF A$<"1" OR A$>"5" THEN 790
810 C=VAL[A$]
820 PRINT "    OK"C
830 PRINT "HOW MANY OF"C"TO BUY";
840 INPUT N
850 IF N*S[C]>BB THEN PRINT "YOU DON'T HAVE ENOUGH MONEY!":G
OTO 840
860 REM ************************
870 REM ADJUST FIGURES AFTER BUY
880 BB=BB-S[C]*N
890 N[C]=N[C]+N
900 TV=0
910 FOR C=1 TO 5
920 TV=TV+N[C]*S[C]
930 NEXT C
940 TV=TV+BB
950  PROCpicture
960 REM ************************
970 REM  MODIFY ALL INDICATORS
980 TV=0
990 FOR C=1 TO 5
1000 K=RND[100]
1010 IF K<P[C] THEN S[C]=S[C]*[1+[P[C]/1000]]
1020 K=RND[100]
1030 IF K<D[C] THEN S[C]=S[C]/[1+[D[C]/1000]]
1040 TV=TV+[S[C]*N[C]]
1050 NEXT C
1060 TV=TV+BB
1070 QQ=QQ*1.005
1080 W=[TV*100/QQ]-100
1090 IF W=0 THEN W=.1
1100 W=W+6
1110 IF W<1 THEN W=1
1120 IF W>15 THEN W=15
1130 RESTORE
```

```
1140 FOR T=1 TO W
1150 READ A$
1160 NEXT T
1170 PRINT
1180 REM ***************************
1190 REM GIVE RATING, START NEW ROUND
1200 PRINT "YOUR RATING AFTER THAT ROUND OF"
1210 PRINT "TRADING IS '";A$;"'"
1220 PRINT:PRINT "   <PRESS SPACEBAR TO CONTINUE >"
1230 A$=GET$
1240 DAY=DAY+1
1250 GOTO 160
1260 DATA "HOPELESS","VERY, VERY POOR"
1270 DATA "TERRIBLE","AWFUL","BAD"
1280 DATA "VERY ORDINARY","AVERAGE"
1290 DATA "REASONABLE","A LITTLE ABOVE AVERAGE"
1300 DATA "FAIRLY GOOD","GOOD","VERY GOOD"
1310 DATA "GREAT","EXCELLENT","SUPERLATIVE"
```

Detroit City

```
10 REM DETROIT CITY - BBC version
20 GOSUB 1640:REM INITIALIZE
30 GOTO 110
40 MT=MT+1:REM COUNTS MONTHS
50 GOSUB 650
60 IF TP>200 THEN 1560
70 PRINT "DO YOU WANT TO RESIGN (Y/N)?"
80 GOSUB 1010
90 IF A$="Y" THEN PRINT "OK, CHIEF":END
100 GOSUB 1380
110 GOSUB 650
120 FOR T=1 TO 1000:NEXT T
130 GOSUB 850
140 PRINT "DO YOU WANT TO EXPAND OUTPUT (Y/N)?"
150 GOSUB 1010
160 IF A$="Y" THEN 1080
170 IF SF=1 THEN 210
180 PRINT "DO YOU WANT TO SELL FACTORY 4 (Y/N)?"
190 GOSUB 1010
200 IF A$="Y" THEN 1250
210 GOSUB 650
220 INPUT "HOW MANY EMPLOYEES TO HIRE";HE
230 NE=NE+HE:IF HE>0 THEN 260
240 INPUT "HOW MANY EMPLOYEES TO FIRE";HE
250 NE=NE-HE
260 GOSUB 650
270 P1=AS:REM SET P1 EQUAL TO OLD PRICE
280 INPUT "WHAT IS YOUR SELLING PRICE";AS
290 REM NEXT LINE REJECTS TOO BIG A CHANGE IN SELLING PRICE
300 IF ABS(P1-AS)>2500 THEN PRINT "TOO BIG A CHANGE FOR THE
MARKET":GOTO 280
310 CLS
320 PRINT:PRINT:PRINT
330 MI=RND(4000)+48*1000:REM THIS MONTH'S SALES BY INDUSTRY
340 C=C+1:REM COUNTS NUMBER OF MONTHS
350 IF C<3 THEN 470
360 M=INT(RND(1)*10+1)/4:REM INFLATION
370 CLS
380 PRINT "INFLATION RATE THIS QUARTER IS"M"%"
```

```
390 PRINT "AVERAGE WAGES BILL WILL NOW RISE TO"
400 AW=(AW*M/100)+AW
410 PRINT TAB(8);"$"INT(AW)"  PER EMPLOYEE"
420 IF INKEY$<>"" THEN 420
430 PRINT:PRINT TAB(12);"ANY KEY TO CONTINUE"
440 IF INKEY$="" THEN 440
450 FA=(FA*M/100)+FA
460 C=0
470 Y(1)=NE*15/12:REM SALES BASED ON NUMBER OF EMPLOYEES
480 Y(2)=(100-AS/FA)*MI/100:REM      SALES BASED ON MONTHLY
INDUSTRY SALES
490 REM    NEXT LINES SET LOWEST FIGURE FROM Y(1),Y(2),M(5)
EQUAL TO Y(3)
500 IF Y(1)<Y(2) AND Y(1)<M(5) THEN Y(3)=Y(1):GOTO 540
510 IF Y(2)<Y(1) AND Y(2)<M(5) THEN Y(3)=Y(2):GOTO 540
520 Y(3)=M(5)
530 REM NEXT LINES DETERMINE MONTHLY SALES
540 IF ABS(P1-AS)<501 THEN Y(3)=3.6*Y(3)/3
550 IF Y(3)>M(5) THEN Y(3)=Y(3)-1975:GOTO 550
560 MC=(MC*M/100)+MC
570 EF=Y(3)/M(5)*100:REM   EFFICIENCY % AS SALES DIVIDED BY
TOTAL OUTPUT
580 AC=(MC*(ABS(85-EF)/3)/100)+MC:REM      AVERAGE COST PER
VEHICLE
590 MP=((Y(3)*(AS-AC))-(NE*AW/12)):REM MONTHLY PROFIT
600 MP=INT(MP/(100*1000))
610 TP=TP+MP/10:REM TOTAL PROFIT IN MILLIONS
620 M=0
630 GOTO 40
640 REM ****************************
650 REM REPORT PRINTOUT
660 CLS
670 PRINT "INDUSTRY SALES"MI"IN MONTH"MT
680 IF MT>0 THEN PRINT "YOUR SALES:"INT(Y(3))" ("INT(Y(3)*10
00/MI)/10"% OF TOTAL )"
690 PRINT "_____________________________________"
700 PRINT "YOU HAVE"NE"EMPLOYEES"
710 PRINT "AVERAGE WAGES ARE $"AW
720 PRINT " OR $M"INT(AW*NE/(100*1000)/12)/10"PER MONTH"
730 PRINT "_____________________________________"
740 IF MT=0 THEN RETURN
```

```
750 PRINT "AVERAGE COST PER VEHICLE  IS $"INT(AC)
760 PRINT "AND AVERAGE SELLING PRICE IS $"INT(AS)
770 PRINT "SO THE AVERAGE PROFIT IS $"INT(AS-AC)
780 PRINT "OR $M"INT((AS-AC)*Y(3)/(100*1000))/10"PER MONTH"
790 PRINT "-----------------------------------"
800 PRINT "PROFIT FOR THE MONTH IS $M"MP/10
810 PRINT "& TOTAL PROFIT TO DATE IS $M"INT(TP*10)/10
820 PRINT "-----------------------------------"
830 RETURN
840 REM ***************************
850 REM MONTH REPORT
860 CLS
870 IF MT>0 THEN PRINT "YOUR MONTHLY SALES ARE"INT(Y(3))
880 PRINT "-----------------------------------"
890 PRINT "MAXIMUM MONTHLY OUTPUT:"
900 PRINT TAB(3);"FACTORY 1:"INT(M(1))
910 PRINT TAB(3);"FACTORY 2:"INT(M(2))
920 PRINT TAB(3);"FACTORY 3:"INT(M(3))
930 IF SF=1 THEN 960
940 PRINT TAB(3);"FACTORY 4:"INT(M(4))
950 PRINT "-----------------------------------"
960 PRINT "TOTAL OUTPUT IS"INT(M(5))
970 PRINT "-----------------------------------"
980 PRINT "EFFICIENCY LEVEL IS"INT(EF)"%"
990 RETURN
1000 REM ***************************
1010 REM GET REPLIES
1020 A$=INKEY$
1030 IF A$<>"Y" AND A$<>"N" THEN 1020
1040 PRINT TAB(22);A$
1050 FOR J=1 TO 500:NEXT J
1060 RETURN
1070 REM ***************************
1080 REM INCREASE OUTPUT?
1090 IF M(4)=0 THEN X=15:GOTO 1110
1100 X=18
1110 PRINT "IT WILL COST $M"X" TO EXPAND"
1120 PRINT TAB(8);"OUTPUT BY 1%"
1130 PRINT "-----------------------------------"
1140 PRINT "HOW MANY % DO YOU WISH TO RAISE OUTPUT?"
1150 INPUT EP:IF EP<0 OR EP>100 THEN 1150
```

```
1160 M(5)=0
1170 FOR T=1 TO 4
1180 M(T)=M(T)+M(T)*EP/100
1190 M(5)=M(5)+M(T)
1200 NEXT T
1210 TP=TP-EP*X
1220 FOR T=1 TO 500:NEXT T
1230 GOTO 170
1240 REM ****************************
1250 REM SALE OF FACTORY FOUR
1260 PRINT "FACTORY 4 IS VALUED FOR SALE AT $M104"
1270 PRINT "YOU CAN'T REBUY IT LATER IF                YOU SELL
 IT..."
1280 PRINT "DO YOU WANT TO SELL (Y/N)?"
1290 GOSUB 1010
1300 IF A$="N" THEN 210
1310 TP=TP+104
1320 SF=1
1330 M(5)=M(1)+M(2)+M(3)
1340 M(4)=0
1350 GOTO 170
1360 REM ****************************
1370 REM CHECK ON LOSSES
1380 IF MP>0 THEN SA=0:GOTO 1480
1390 SA=SA+1
1400 IF SA>11 THEN 1420
1410 GOTO 1480
1420 CLS:PRINT
1430 PRINT "YOU JUST MADE YOUR TWELFTH MONTHLY"
1440 PRINT "LOSS IN A ROW....................."
1450 PRINT TAB(6);"YOUR EMPLOYMENT"
1460 PRINT TAB(6);"IS HEREBY TERMINATED!!"
1470 END
1480 IF TP>=-250 THEN 1530
1490 CLS:PRINT
1500 PRINT "UNDER YOUR MANAGEMENT, THE COMPANY HAS"
1510 PRINT "LOST MORE THAN $M250!"
1520 GOTO 1450
1530 IF TP>200 THEN 1570
1540 RETURN
1550 REM ****************************
```

```
1560 REM SWEET SWEET SUCCESS!!!
1570 CLS:PRINT
1580 PRINT "WELL DONE! THE COMPANY HAS MADE MORE"
1590 PRINT "   THAN $M200.    YOU'VE BEEN MADE"
1600 PRINT "         A MEMBER OF THE BOARD"
1610 FOR T=1 TO 2000:NEXT T
1620 END
1630 REM ****************************
1640 REM INITIALIZATION
1650 PRINT RND(TIME)
1660 MODE 6:VDU 19,0,3;0;19,1,1;0;
1670 DIM M(5),Y(5)
1680 NE=12000:REM STARTING NO EMPLOYEES
1690 AW=22995:REM STARTING AVERAGE WAGE
1700 AC=11100:REM COST PRICE/VEHICLE
1710 AS=12000:REM SELLING PRICE
1720 MI=50*1000:MC=10100
1730 Y(3)=12500
1740 MS=25:EF=77:FA=160:SF=0:MT=0
1750 FOR J=1 TO 5
1760 READ M(J)
1770 NEXT J
1780 RETURN
1790 DATA 8900,3250,2500,1625,16275
```

Gridiron

```
10 REM GRIDIRON
20 PRINT RND(TIME)
30 MODE 6:VDU 19,0,2;0;19,1,7;0;
40 GOSUB 70
50 GOTO 220
60 REM ****************************
70 FOR X=1 TO 1500:NEXT X
80 RETURN
90 REM ****************************
100 PRINT A$;SA
110 PRINT B$;SB
120 RETURN
130 REM ****************************
140 IF Z$=A$ THEN Z$=B$:RETURN
150 Z$=A$:RETURN
160 REM ****************************
170 IF INKEY$<>"" THEN 170
180 PRINT "      > PRESS ANY KEY <"
190 IF INKEY$="" THEN 190
200 PRINT TAB(20);"OK":RETURN
210 REM ****************************
220 REM INITIALIZE
230 DEF FNA(X)=INT(RND(1)*X)+1
240 PRINT "ONE PLAYER OR TWO"
250 INPUT X
260 IF X<1 OR X>2 THEN 250
270 IF X=1 THEN VC=1:A$="SILICON COWBOYS":GOTO 300
280 PRINT "WHAT IS THE NAME OF THE HOME TEAM?"
290 INPUT A$:IF A$="" THEN 290
300 PRINT "AND THE NAME OF THE VISITING TEAM?"
310 INPUT B$:IF B$="" THEN 310
320 Z$=A$:NU=35
330 CLS
340 PRINT "THERE ARE"INT(10*(60-(W/4)))/10"MINUTES TO GO"
350 PRINT TAB(8);Z$" TO KICK OFF"
360 PRINT "YOU ARE ON YOUR OWN"NU"YARD LINE"
370 IF VC=1 AND Z$=A$ THEN GOSUB 70:GOTO 400
380 PRINT "TO KICK OFF..."
390 GOSUB 170
```

```
400 A=FNA[20]+40
410 PRINT Z$;" HAVE..."
420 FOR X=1 TO A
430 PRINT TAB[X/3];"KICKED"X"YARDS"
440 NEXT X
450 NU=NU+X
460 GOSUB 70
470 PRINT "THE BALL IS CAUGHT!"
480 GOSUB 70
490 A=FNA[30]+10
500 FOR X=1 TO A
510 PRINT TAB[X/5];"AND RETURNED"X"YARDS"
520 NEXT X
530 NU=ABS[100-NU+X]
540 GOSUB 140
550 PRINT "_____"
560 PRINT "THE BALL IS DOWN ON"
570 PRINT Z$"'S"NU"YARD LINE"
580 IF Z$=A$ AND VC=1 THEN GOSUB 70:GOTO 600
590 GOSUB 170
600 TG=10:D=0:SL=NU
610 IF W=60 OR W=180 THEN 2010
620 IF W=120 THEN 2070
630 IF W=240 THEN 2140
640 CLS
650 PRINT LEFT$[A$,6];SA;LEFT$[B$,6];SB
660 PRINT INT[10*[60-[W/4]]]/10"MINUTES TO GO"
670 GOSUB 70
680 PRINT "_____"
690 PRINT Z$" IN POSSESSION"
700 PRINT TAB[4];D"DOWN"
710 PRINT TAB[4];TG"YARDS TO GO"
720 PRINT "_____"
730 PRINT "START AT"SL"YARD LINE"
740 PRINT "NOW ON"NU"YARD LINE"
750 PRINT 100-NU"YARDS TO TOUCHDOWN"
760 PRINT "_____"
770 PRINT "ON THIS PLAY ";
780 IF Z$=A$ THEN PRINT A$;" CAN":GOTO 800
790 PRINT B$;" CAN"
800 PRINT "EITHER 1 - THROW"
```

```
810 PRINT "        2 - CARRY"
820 PRINT "    OR 3 - PUNT"
830 P=0
840 IF Z$=A$ AND VC=1 AND D<3 THEN P=2:GOTO 900
850 IF Z$=A$ AND VC=1 AND TG<7 THEN P=2:GOTO 900
860 IF Z$=A$ AND VC=1 AND (100-NU)<31 THEN P=3:GOTO 900
870 IF Z$=A$ AND VC=1 THEN P=1:GOTO 900
880 K$=INKEY$:IF K$<"1" OR K$>"3" THEN 880
890 P=VAL(K$):PRINT TAB(10);"OK"P
900 GOSUB 70
910 W=W+1
920 CLS
930 PRINT Z$;", YOUR QUARTERBACK HAS"
940 PRINT TAB(8);"GOT THE BALL"
950 PRINT "_____________________________________"
960 PRINT "WAIT FOR THE COUNT, ";Z$;","
970 PRINT TAB(8);"THEN HIT ANY KEY..."
980 IF INKEY$<>"" THEN 980
990 GOSUB 70
1000 GOSUB 2200
1010 IF E=11 THEN 2340
1020 PRINT "_____________________________________"
1030 ON P GOTO 1050,1310,1590
1040 REM **************************
1050 PRINT "YOU'VE THROWN"E*5"YARDS"
1060 PRINT TAB(4);"AND THE PLAY IS..."
1070 A=FNA(8)
1080 IF A=1 THEN 1520
1090 A=FNA(E+1)
1100 IF A=1 THEN PRINT TAB(20);"...COMPLETE":GOTO 1220
1110 PRINT TAB(20);"...INCOMPLETE":D=D+1
1120 GOSUB 170
1130 PRINT "_____________________________________"
1140 IF D>3 THEN 1160
1150 GOTO 610
1160 PRINT "THAT WAS YOUR 4TH DOWN"
1170 PRINT "AND YOU'VE LOST POSSESSION!!"
1180 D=0:TG=10:NU=ABS(100-NU):SL=NU
1190 GOSUB 70
1200 GOSUB 140
1210 GOTO 610
```

```
1220 GOSUB 170
1230 NU=NU+(E*5):TG=TG-(E*5)
1240 IF NU>100 THEN 1800
1250 IF TG<1 THEN 1280
1260 D=D+1:IF D>3 THEN 1160
1270 GOTO 610
1280 D=0:TG=10:SL=NU
1290 GOTO 610
1300 REM ***************************
1310 A=FNA(15)
1320 IF A=1 THEN 1510
1330 E=A-5
1340 IF E<0 THEN 1440
1350 IF E=0 THEN E=1:GOTO 1370
1360 PRINT "GOOD SNAP, PASS AND RUN"
1370 PRINT "YOU'VE GAINED"E"YARDS"
1380 GOSUB 170
1390 TG=TG-E:NU=ABS(NU+E):D=D+1
1400 IF NU>100 THEN 1800
1410 IF TG<1 THEN 1280
1420 IF D>3 THEN 1160
1430 GOTO 610
1440 PRINT "GREAT RUNNING BY THE OPPOSITION HAS"
1450 PRINT "CAUSED YOU TO LOSE"ABS(E)"YARDS"
1460 TG=TG-E:NU=NU+E:D=D+1
1470 GOSUB 170
1480 IF D>3 THEN 1160
1490 GOTO 610
1500 REM ***************************
1510 PRINT "BAD SNAP...YOU'VE"
1520 PRINT "FUMBLED...AND"
1530 PRINT "YOU'VE LOST POSSESSION..."
1540 NU=100-NU:D=0:TG=10:SL=NU
1550 REM ***************************
1560 GOSUB 170
1570 GOTO 460
1580 REM ***************************
1590 PRINT "NICE PUNT..."
1600 PRINT "YOU'VE KICKED"E*4"YARDS"
1610 NU=NU+E*4
1620 IF NU>100 THEN 1650
```

```
1630 PRINT "_____"
1640 GOTO 460
1650 A=FNA(3)
1660 IF A>1 THEN 1740
1670 PRINT "BUT YOU'VE MISSED THE GOAL!!"
1680 IF NU-E*4<80 THEN NU=ABS(100-(NU-E*4)):GOTO 1700
1690 NU=20
1700 D=0:TG=10:SL=NU
1710 GOSUB 140
1720 GOSUB 170
1730 GOTO 610
1740 PRINT ".....AND SCORED!"
1750 IF Z$=B$ THEN SB=SB+3:GOTO 1770
1760 SA=SA+3
1770 GOSUB 100
1780 GOSUB 170
1790 NU=35:GOTO 330
1800 CLS
1810 FOR X=1 TO 5
1820 PRINT TAB(X*2);"TOUCHDOWN!!!"
1830 NEXT X
1840 IF Z$=A$ THEN SA=SA+6:GOTO 1860
1850 SB=SB+6
1860 GOSUB 100
1870 PRINT "TO PLAY FOR EXTRA POINT"
1880 GOSUB 170
1890 PRINT "_____"
1900 PRINT "THE BALL IS SNAPPED...PREPARE TO KICK!"
1910 GOSUB 70
1920 GOSUB 2200
1930 IF E>9 THEN PRINT "YOU MISSED":NU=20:GOTO 1970
1940 PRINT "YOU SCORED...":NU=35
1950 IF Z$=A$ THEN SA=SA+1:GOTO 1980
1960 SB=SB+1:GOTO 1980
1970 GOSUB 140
1980 GOSUB 100
1990 GOSUB 170
2000 GOTO 330
2010 FOR X=1 TO 10
2020 PRINT TAB(2*X);"PERIOD OVER"
2030 NEXT X
```

```
2040 GOSUB 100
2050 GOSUB 170
2060 GOTO 660
2070 FOR X=1 TO 10
2080 PRINT TAB(2*X);"HALF TIME"
2090 NEXT X
2100 GOSUB 100
2110 Z$=B$
2120 GOSUB 170
2130 NU=35:W=W+2:GOTO 330
2140 FOR X=1 TO 10
2150 PRINT TAB(2*X);"GAME OVER"
2160 NEXT X
2170 GOSUB 100
2180 END
2190 REM ***************************
2200 E=0:X=10
2210 IF Z$=A$ AND VC=1 THEN PRINT "THIS ONE FOR ";A$:GOTO 22
90
2220 E=E+1:X=X-1
2230 PRINT TAB(E);E
2240 FOR Y=1 TO X*1.5
2250 IF INKEY$<>"" THEN Y=X*1.5+1:RETURN
2260 NEXT Y
2270 IF E=11 THEN RETURN
2280 GOTO 2220
2290 FOR E=1 TO FNA(7)+2
2300 FOR J=1 TO 60:NEXT J
2310 PRINT TAB(E);E
2320 NEXT E
2330 RETURN
2340 PRINT "TOO LATE!"
2350 PRINT "YOU'VE BEEN SACKED!"
2360 E=FNA(4)
2370 IF E=3 THEN 2430
2380 PRINT "AND LOST FIVE YARDS!"
2390 TG=TG+5:D=D+1:NU=NU-5
2400 GOSUB 170
2410 IF D>3 THEN 1160
2420 GOTO 610
2430 PRINT "AND LOST POSSESSION!"
```

```
2440 D=0:NU=ABS(100-NU+5):SL=NU:TG=10
2450 GOSUB 170
2460 GOSUB 140
2470 GOTO 610
```

Tennis

```
10 REM TENNIS - BBC version
20 PRINT RND(TIME)
30 MODE 6:VDU 19,0,3;0;19,1,1;0;
40 AA=0:BB=0:T=0:KA=0
50 XA=0:YA=0:ZA=0
60 XB=0:YB=0:ZB=0
70 DEF FNA(X)=INT(RND(1)*X)+1
80 INPUT "ONE HUMAN PLAYER OR TWO";A
90 IF A<1 OR A>2 THEN 80
100 IF A=1 THEN A$="BJORNX":VC=1
110 IF VC=1 THEN 160
120 PRINT "PLEASE ENTER A SIX-LETTER NAME"
130 INPUT "NAME OF FIRST PLAYER";A$
140 IF LEN(A$)<6 THEN A$=A$+" ":GOTO 140
150 A$=LEFT$(A$,6)
160 INPUT "NAME OF SECOND PLAYER";B$
170 IF LEN(B$)<6 THEN B$=B$+" ":GOTO 170
180 B$=LEFT$(B$,6)
190 S=1:AA=1:BB=1
200 CLS
210 P$=A$:R$=B$
220 REM ************************************
230 IF P$=A$ THEN R$=B$
240 IF P$=B$ THEN R$=A$
250 PRINT P$;" SERVING"
260 PRINT "DO YOU WANT TO SERVE 1 - FAST"
270 PRINT "                      OR 2 - SLOW"
280 IF P$=A$ AND VC=1 AND SC=0 THEN KB=1:GOSUB 1720:GOTO 330

290 IF P$=A$ AND VC=1 AND SC=1 THEN KB=2:GOSUB 1720:GOTO 330

300 K$=INKEY$
310 IF K$<"1" OR K$>"2" THEN 300
320 KB=VAL(K$)
330 PRINT:PRINT TAB(6);KB;TAB(10);"> IT'S A ";
340 IF KB=1 THEN PRINT "FAST";
350 IF KB=2 THEN PRINT "SLOW";
360 PRINT " SERVE..."
370 GOSUB 1720
```

```
380 IF KB=1 THEN EB=FNA(3):GOTO 400
390 EB=FNA(8)
400 IF EB=1 THEN 450
410 IF EB=3 AND SC=0 THEN 520
420 IF EB=3 AND SC=1 THEN 590
430 GOTO 670
440 REM ************************************
450 CLS:PRINT
460 PRINT TAB(8);"....ACE...."
470 GOSUB 1720
480 SC=0
490 IF P$=A$ THEN 1140
500 GOTO 1150
510 REM ************************************
520 CLS:PRINT
530 PRINT TAB(12);"....OUT...."
540 PRINT TAB(8);"...SECOND SERVE..."
550 GOSUB 1720
560 SC=1
570 GOTO 230
580 REM ************************************
590 CLS:PRINT
600 PRINT TAB(12);"....OUT...."
610 PRINT TAB(9);"....DOUBLE FAULT..."
620 GOSUB 1720
630 SC=0
640 IF P$=A$ THEN 1150
650 GOTO 1140
660 REM ************************************
670 SC=0
680 CLS:PRINT
690 IF INKEY$<>"" THEN 690
700 PRINT R$;", THE BALL IS":PRINT "IN YOUR COURT"
710 PRINT "_____________________________"
720 IF R$=A$ AND VC=1 THEN 750
730 PRINT "HIT ANY KEY, WHEN YOU SEE THE ZERO,          TO R
ETURN THE BALL..."
740 IF INKEY$<>"" THEN 740
750 X=4*FNA(3):Y=X
760 GOSUB 1720
770 E=5
```

```
780 PRINT TAB(2*(11-E));E
790 Y=Y-1
800 S$=INKEY$
810 IF S$<>"" AND E=0 THEN 890
820 IF S$<>"" THEN 990
830 IF Y>0 THEN 790
840 E=E-1:Y=X
850 IF E<-1 THEN 990
860 IF E=-1 AND R$=A$ AND VC=1 THEN 890
870 GOTO 780
880 IF KB=1 THEN EA=FNA(2):GOTO 1000
890 EA=FNA(4)
900 IF E=0 AND R$=A$ AND VC=1 THEN EA=FNA(8)
910 IF EA=1 THEN 940
920 IF R$=A$ THEN R$=B$:GOTO 670
930 R$=A$:GOTO 670
940 PRINT R$;", YOU'VE HIT THE BALL"
950 PRINT TAB(8);"OUT OF PLAY..."
960 GOSUB 1720
970 IF R$=A$ THEN R$=B$:GOTO 1150
980 GOTO 1140
990 EA=FNA(4)
1000 IF EA=1 THEN 1070
1010 PRINT "YOU MISSED THE BALL, AND..."
1020 GOSUB 1720
1030 PRINT "   IT WAS IN...BAD MISTAKE"
1040 GOSUB 1720
1050 IF R$=A$ THEN R$=B$:GOTO 1150
1060 GOTO 1140
1070 PRINT "YOU MISSED THE BALL AND..."
1080 GOSUB 1720
1090 PRINT "   IT WAS OUT..WELL LEFT"
1100 GOSUB 1720
1110 IF R$=A$ THEN R$=B$:GOTO 1140
1120 GOTO 1150
1130 REM ********************************
1140 AA=AA+1:GOTO 1160
1150 BB=BB+1
1160 IF AA<5 AND BB<5 THEN 1230
1170 IF (BB>4 AND AA<4) OR (BB>4 AND BB-AA>1) THEN AA=1:BB=1
:GOTO 1500
```

```
1180 IF (AA>4 AND BB<4) OR (AA>4 AND AA-BB>1) THEN AA=1:BB=1
:GOTO 1440
1190 IF AA>4 AND AA>BB THEN C$="ADV":D$="---":GOTO 1320
1200 IF BB>4 AND BB>AA THEN D$="ADV":C$="---":GOTO 1320
1210 C$="{DEUCE":D$="{DEUCE":GOTO 1320
1220 REM ********************************
1230 RESTORE
1240 FOR D=1 TO AA
1250 READ C$
1260 NEXT D
1270 RESTORE
1280 FOR D=1 TO BB
1290 READ D$
1300 NEXT D
1310 REM ********************************
1320 CLS
1330 PRINT "_____________________________"
1340 PRINT "              SET SET SET"
1350 PRINT "_____________________________"
1360 PRINT "                1   2   3  GAME"
1370 PRINT A$;"      ";XA;" ";YA;" ";ZA;" ";C$
1380 PRINT B$;"      ";XB;" ";YB;" ";ZB;" ";D$
1390 PRINT "_____________________________"
1400 GOSUB 1720
1410 IF T<>1 THEN 230
1420 END
1430 REM ********************************
1440 CLS
1450 PRINT "GAME TO ";A$
1460 GOSUB 1720
1470 IF S=1 THEN XA=XA+1:C$="0":D$="0":GOTO 1560
1480 IF S=2 THEN YA=YA+1:C$="0":D$="0":GOTO 1580
1490 IF S=3 THEN ZA=ZA+1:C$="0":D$="0":GOTO 1600
1500 CLS
1510 PRINT "GAME TO ";B$
1520 GOSUB 1720
1530 IF S=1 THEN XB=XB+1:C$="0":D$="0":GOTO 1560
1540 IF S=2 THEN YB=YB+1:C$="0":D$="0":GOTO 1580
1550 IF S=3 THEN ZB=ZB+1:C$="0":D$="0":GOTO 1600
1560 IF (XA>5 AND XB<5) OR (XA<5 AND XB>5) THEN 1630
1570 IF (XA>5 AND XA-XB>1) OR (XB>5 AND XB-XA>1) THEN 1630
```

```
1580 IF (YA>5 AND YB<5) OR (YA<5 AND YB>5) THEN 1630
1590 IF (YA>5 AND YA-YB<1) OR (YB>5 AND YB-YA>1) THEN 1630
1600 IF (ZA>5 AND ZB<5) OR (ZA<5 AND ZB>5) THEN 1680
1610 IF (ZA>5 AND ZA-ZB>1) OR (ZB>5 AND ZB-ZA>1) THEN 1680
1620 GOTO 1640
1630 S=S+1
1640 AA=1:BB=1
1650 IF P$=A$ THEN R$=A$:P$=B$:GOTO 1320
1660 P$=A$:R$=B$:GOTO 1320
1670 REM ***********************************
1680 T=1
1690 GOTO 1320
1700 REM ***********************************
1710 REM DELAY
1720 FOR M=1 TO 1000:NEXT M
1730 RETURN
1740 DATA "0","15","30","40"
```

Grand Prix

```
10 REM GRAND PRIX
20 GOSUB 2210:REM Initialize
30  PROCtrack
40 REM ****************************
50 REM MAJOR LOOP
60 REPEAT
70  PROCprintout
80  PROCaccel
90  PROCbrakes
100  PROCcorner
110 UNTIL FALSE
120 REM ****************************
130  DEFPROCprintout
140 CLS
150 PRINT "ENGINE TEMPERATURE"INT(ENG*10)/10"C.  (MAX. 200)"
160 PRINT "BRAKE TEMPERATURE:"INT(BRAK*10)/10"C.  (MAX. 500)"
170 PRINT " DISTANCE COVERED:"INT(DIST*10)/10"METERS"
180 PRINT "                  :"INT(DIST*100/RR)/100"LAPS"
190 PRINT "YOU'RE IN POSITION"INT(FP)
200 PRINT "-----------------------------------------"
210 PRINT "    CURRENT SPEED:"INT(SPEED*10)/10"KPH"
220 PRINT "                  :"INT(SPEED*5.555)/10"METERS PER M
OVE"
230 PRINT "-----------------------------------------"
240 PRINT "CORNER APPROACHING"INT(APP)"METERS"
250 PRINT "RECOMMENDED SPEED:"C(C)"KPH"
260 PRINT "-----------------------------------------"
270  ENDPROC
280 REM ****************************
290  DEFPROCaccel
300 X$=INKEY$:IF X$<>"Z" AND X$<>"M" AND X$<>" " THEN 300
310 PRINT TAB(12);"OK"
320 X=0
330 IF X$="M" THEN X=SPEED/15
340 IF X$="Z" THEN X=-SPEED/15
350 NUM=NUM+1:REM NUMBER OF MOVES
360 SPEED=SPEED+X
370 IF SPEED<0 THEN SPEED=0
380 TRAV=SPEED*.5555:REM DISTANCE
  TRAVELLED
```

```
390 DIST=DIST+TRAV:REM TOTAL DISTANCE
  TRAVELLED
400 ENG=ENG+(X/2)+.07:IF ENG<70 THEN ENG=70+RND(1)*8:REM ENGIN
E TEMP
410 IF X>0 THEN BRAK=BRAK*.9:REM BRAKE    TEMP FALLING; ACCELE
RATING
420 IF X<1 THEN BRAK=BRAK-(3*X)-RND(1)*3:REM BRAKE TEMP INCREA
SING; BRAKING
430 IF BRAK<8 THEN BRAK=8+RND(1)*8
440  ENDPROC
450 REM ****************************
460  DEFPROCbrakes
470 IF ENG>200 THEN PRINT "YOUR ENGINE HAS OVERHEATED":GOTO 84
0
480 IF BRAK>500 THEN PRINT "YOUR BRAKES HAVE OVERHEATED":GOTO
840
490  ENDPROC
500 REM ****************************
510  DEFPROCcorner
520 APP=APP-TRAV
530 IF APP>0 THEN RETURN
540 CRASH=0
550 IF SPEED>(C(C)*1.125) THEN CRASH=1:GOTO 700
560 IF SPEED>(C(C)*1.1) THEN GOTO 700
570 PNT=PNT+100-((C(C)*1.1)-SPEED):REM CORNERING POINTS
580 NC=NC+1:REM NUMBER OF CORNERS
590 CP=96-(PNT/NC):REM CORNERING POSITION
600 AM=AM+A(C):REM AVERAGE NUMBER OF MOVES ALLOWED
610 RP=NUM-AM:REM RACING POSITION: YOUR MOVES MINUS AVERAGE MO
VES
620 FP=(CP+RP)/2:REM FIELD POSITION IS  AVERAGE OF CORNER & RA
CE POSITIONS
630 IF FP<1 THEN FP=1
640 C=C+1
650 IF C=WW THEN C=1:REM LAP OVER
660 APP=APP+D(C)
670 IF LAP*QQ=AM THEN 920:REM RACE OVER
680  ENDPROC
690 REM ****************************
700 REM CRASHED
710 CLS
```

```
720 PRINT "YOU CORNERED AT"INT(10*SPEED)/10"KPH"
730 PRINT "AND THE MAXIMUM SPEED WAS JUST"C(C)
740 GOSUB 2340
750 PRINT "YOU SPIN OFF THE TRACK...
760 GOSUB 2340
770 IF CRASH=1 THEN 840
780 PRINT "YOU'VE LOST 20 SECONDS, BUT YOU'RE      ABLE TO REJ
OIN THE RACE"
790 NUM=NUM+10:SPEED=INT(2*C(C)/3)
800 PNT=PNT+50
810 GOSUB 2340
820 GOTO 580
830 REM ****************************
840 PRINT ".....AND CRASH!!!!!"
850 PRINT "_____________________________________"
860 PRINT "YOU ONLY COMPLETED"INT(10*DIST)/10"METERS,"
870 PRINT "OR"INT(DIST*100/RR)/100"LAPS AND AT THAT"
880 PRINT "STAGE WERE IN POSITION"INT(FP)
890 PRINT "_____________________________________"
900 GOTO 1060
910 REM ****************************
920 REM RACE OVER
930 CLS
940 EFLAG=1
950 FOR X=1 TO 20
960 PRINT TAB(X);"WELL DONE, ";A$;"!!"
970 PRINT TAB(21-X);"WELL DONE, ";A$;"!!"
980 NEXT X
990 PRINT "_____________________________________"
1000 PRINT "YOU MANAGED TO LAST OUT THE FULL                "
LAP"LAP RACE..."
1010 PRINT "_____________________________________"
1020 GOSUB 2340
1030 PRINT "YOU FINISHED IN POSITION"INT(FP)
1040 PRINT "AFTER STARTING IN 6TH POSITION..."
1050 GOSUB 2340
1060 PRINT "YOUR AVERAGE SPEED WAS"INT(DIST*180/NUM)/100"KPH"
1070 GOSUB 2340
1080 IF RP<1 THEN RP=1
1090 IF CP<1 THEN CP=1
1100 PRINT "YOU WERE"INT(ABS(RP))"TH FASTEST ON STRAIGHTS,"
```

```
1110 PRINT "AND"INT[ABS[CP]]"TH FASTEST ON CORNERS."
1120 PRINT:PRINT "PRESS 'S' FOR SAME RACE, 'N' FOR NEW     RAC
E, 'E' TO END"
1130 I$=INKEY$:IF I$<>"S" AND I$<>"N" AND I$<>"E" THEN 1130
1140 IF I$="E" THEN END
1150 GOSUB 2250
1160 RESTORE
1170 IF I$="S" THEN GOSUB 1500:LAP=L2AP:GOTO 70
1180 IF I$="N" THEN CLS:GOSUB 1260:GOTO 70
1190 REM ***************************
1200  DEFPROCtrack
1210 INPUT "WHAT IS YOUR NAME, DRIVER";A$
1220 PRINT
1230 FOR X=1 TO 3
1240 PRINT TAB[4*X];"OK, GOOD LUCK, ";A$
1250 GOSUB 2340:NEXT X
1260 PRINT "****************************************"
1270 PRINT "WHICH RACE DO YOU WANT TO TAKE PART IN:"
1280 PRINT
1290 PRINT TAB[7];"BRITISH GRAND PRIX  2650MT  :1"
1300 PRINT TAB[7];"GERMAN  GRAND PRIX  1700MT  :2"
1310 PRINT TAB[7];"ITALIAN GRAND PRIX  2200MT  :3"
1320 PRINT TAB[7];"MONACO  GRAND PRIX  3100MT  :4"
1330 PRINT
1340 PRINT TAB[7];"ENTER A NUMBER [1 TO 4]"
1350 K$=INKEY$
1360 IF K$<"1" OR K$>"4" THEN 1350
1370 GP=VAL[K$]
1380 PRINT "****************************************"
1390 PRINT TAB[8];"OK, THE ";
1400 IF GP=1 THEN PRINT "BRITISH";
1410 IF GP=2 THEN PRINT "GERMAN";
1420 IF GP=3 THEN PRINT "ITALIAN";
1430 IF GP=4 THEN PRINT "MONACO";
1440 PRINT " RACE"
1450 PRINT "****************************************"
1460 PRINT:PRINT "OVER HOW MANY LAPS?"
1470 INPUT LAP:IF LAP<1 THEN 1470
1480 LAP=INT[LAP+.5]:L2AP=LAP
1490 REM ***************************
1500 REM BRITISH DATA
```

```
1510 SPEED=140
1520 FOR X=1 TO 9
1530 READ D[X]:REM DISTANCE BETWEEN CORNERS
1540 NEXT X
1550 DATA 800,400,250,200,250,300,100,100,250
1560 FOR X=1 TO 9
1570 READ C[X]:REM RECOMMENDED MAXIMUM CORNERING SPEED
1580 NEXT X
1590 DATA 150,90,175,200,200,90,90,150,150
1600 FOR X=1 TO 9
1610 READ A[X]:REM AVERAGE NUMBER OF MOVES ALLOWED BETWEEN COR
NERS
1620 NEXT X
1630 DATA 8,4,2,2,2,2,1,1,2
1640 APP=800:WW=10:QQ=24:RR=2650
1650 IF GP=1 THEN ENDPROC
1660 REM **************************
1670 REM GERMAN DATA
1680 SPEED=85
1690 FOR X=1 TO 7
1700 READ D[X]
1710 NEXT X
1720 DATA 600,200,100,150,250,200,200
1730 FOR X=1 TO 7
1740 READ C[X]
1750 NEXT X
1760 DATA 90,175,120,90,200,200,175
1770 FOR X=1 TO 7
1780 READ A[X]
1790 NEXT X
1800 DATA 6,2,1,2,2,2,2
1810 APP=600:WW=8:QQ=17:RR=1700
1820 IF GP=2 THEN ENDPROC
1830 REM **************************
1840 REM ITALIAN DATA
1850 SPEED=108
1860 FOR X=1 TO 7
1870 READ D[X]
1880 NEXT X
1890 DATA 800,300,100,150,300,350,200
1900 FOR X=1 TO 7
```

```
1910 READ C[X]
1920 NEXT X
1930 DATA 120,90,90,150,200,120,150
1940 FOR X=1 TO 7
1950 READ A[X]
1960 NEXT X
1970 DATA 8,3,2,1,3,3,2
1980 APP=800:WW=8:QQ=22:RR=2200
1990 IF GP=3 THEN ENDPROC
2000 REM ***************************
2010 REM MONACO DATA
2020 SPEED=162.5
2030 FOR X=1 TO 14
2040 READ D[X]
2050 NEXT X
2060 DATA 400,100,100,300,400,300,150,200,200,200
2070 DATA 150,150,200,250
2080 FOR X=1 TO 14
2090 READ C[X]
2100 NEXT X
2110 DATA 175,150,175,200,120,200,175,90,175,150
2120 DATA 150,175,120,150
2130 FOR X=1 TO 14
2140 READ A[X]
2150 NEXT X
2160 DATA 4,1,1,3,4,3,1,2,2,2
2170 DATA 1,2,2,2
2180 ENDPROC
2190 RETURN
2200 REM ***************************
2210 REM Initialize
2220 PRINT RND[TIME]
2230 MODE 6:VDU 19,0,3;0;19,1,1;0;
2240 DIM A[14],C[14],D[14]
2250 C=1:FP=6:PNT=0:NC=0:CP=0
2260 AM=0:RP=0:APP=0
2270 NUM=0:REM NUMBER OF MOVES
2280 ENG=100:BRAK=10:TRAV=0:DIST=0
2290 EFLAG=0
2300 X=0
2310 RETURN
2320 REM ***************************
```

```
2330 REM DELAY
2340 FOR O=1 TO 1000:NEXT O
2350 RETURN
```

Flight Simulation

```
10 REM FLIGHT SIMULATION - BBC VERSION
15 REM Engage CAPS LOCK before running
20 RPT=0
30 LD=RND(359)
40 DIM E$(1000):REM THIS HOLDS FLIGHT RECORD
50 DIM A$(7),C$(7):REM THESE ARRAYS HOLD HORIZON AND COMPASS
 OUTPUT
60 REM ******************************
70  PROCinitial
80 IF CRASH=0 THEN GOSUB 820:REM HORIZON/COMPASS
90  PROCpicture
100 IF CRASH=1 THEN END
110 IF LAND=1 AND UFLAG=1 THEN PRINT "WELL DONE. A PERFECT L
ANDING!!":END
120 IF LAND=1 AND UFLAG=0 THEN PRINT "YOUR WHEELS ARE UP":GO
SUB 1780:GOTO 90
130 T=AIRSPEED:STALL=0
140 X$=INKEY$(0)
150 IF X$="R" THEN RPT=1:GOTO 70
160 IF RPT=1 AND E$(CLOCK+1)="" THEN RPT=0:GOTO 140
170 IF RPT=1 THEN X$=E$(CLOCK+1)
180 IF X$="" THEN 140
190 IF CLOCK<999 THEN E$(CLOCK+1)=X$
200 IF TAKEOV=1 THEN ELEVATE=INT(ELEVATE+RND(1)*2-RND(1)*3)
210 IF AIRSPEED<3 THEN 290
220 IF X$="Q" THEN ELEVATE=ELEVATE+5:EFLAG=5:IF ELEVATE>60 T
HEN STALL=1
230 IF X$="A" THEN ELEVATE=ELEVATE-5:EFLAG=-5:IF ELEVATE<-70
 THEN STALL=-1
240 IF STALL<>0 THEN GOSUB 1640
250 IF ALTIMETER<1 THEN 290:REM PREVENTS DRAMATIC TURNS ON T
HE GROUND
260 IF X$="Z" THEN WA=WA-.5:ANG=ANG-6:IF WA<-3 THEN WA=-3
270 IF X$="M" THEN WA=WA+.5:ANG=ANG+6:IF WA>3 THEN WA=3
280 ANG=INT(ANG+RND(1)*2-RND(1)*2)
290 IF X$=" " THEN AIRSPEED=AIRSPEED+8.5
300 IF X$="." THEN AIRSPEED=AIRSPEED-7
310 AIRSPEED=AIRSPEED-ELEVATE/5
320 IF UFLAG=1 THEN AIRSPEED=AIRSPEED-1.5:FUEL=FUEL-.5
```

```
330 IF AIRSPEED<0 THEN AIRSPEED=0
340 IF AIRSPEED>400 THEN AIRSPEED=400
350 IF X$="1" AND UFLAG=0 THEN UFLAG=1:GOTO 370
360 IF X$="1" AND UFLAG=1 THEN UFLAG=0
370 FUEL=FUEL-(ABS(T-AIRSPEED)/10)-3.75
380 IF FUEL<1 THEN GOSUB 1780
390 IF TAKEOV=1 THEN 420
400 IF ELEVATE>10 AND AIRSPEED>45 AND AIRSPEED<60 AND UFLAG=
1 THEN TAKEOV=1
410 IF TAKEOV=0 THEN ALTIMETER=0:GOTO 450
420 IF LAND=0 AND AIRSPEED<30 THEN ELEVATE=ELEVATE-5:ALTIMET
ER=9*ALTIMETER/10
430 ALTIMETER=ALTIMETER+INT(((ELEVATE+.1)*AIRSPEED)+EFLAG*AI
RSPEED/1000)/80
440 IF ALTIMETER<300 AND TAKEOV=1 THEN ALTIMETER=ALTIMETER+A
IRSPEED/30+ELEVATE
450 IF ALTIMETER<0 THEN GOSUB 1780:REM CRASH
460 REM CHANGE NEXT TWO LINES TO MAKE IT EASIER (OR EVEN HAR
DER) TO LAND
470 IF ALTIMETER>15 AND AIRSPEED>20 OR TAKEOV=0 THEN 80
480 IF ABS(ANG-LD)<13 OR ABS(ANG+360-LD)<13 THEN LAND=1:GOTO
 80
490 REM ******************************
500  DEFPROCpicture
510 CLS
520 PRINT "   HORIZON";TAB(20);"HEADING"
530 EV=INT(ELEVATE/10)
540 IF EV>2 THEN EV=2
550 IF EV<-2 THEN EV=-2
560 IF EV<>0 AND TAKEOV=1 AND CRASH=0 THEN GOSUB 1920
570 PRINT ":--------------------:----------:"
580 FOR J=1 TO 7
590 PRINT ": ";A$(J);" :";C$(J);" :"
600 A$(J)=""
610 NEXT J
620 PRINT ":------------------------------------:"
630 DIST=DIST+ABS((COS(ELEVATE))*AIRSPEED)/360
640 CLOCK=CLOCK+1
650 PRINT ":RANGE"INT(DIST*10)/10": TIME"INT(CLOCK)/10;":"LD

660 PRINT ":------------------------------------:"
670 PRINT ":AIRSPEED :"INT(AIRSPEED)
```

```
680 PRINT ":";LEFT$(Q$,INT(AIRSPEED/20));">"
690 PRINT ":ALTIMETER:"INT(ALTIMETER);
700 IF ANG<0 THEN PRINT TAB(19);360+ANG"DEG."
710 IF ANG>=0 THEN PRINT TAB(19);ANG"DEG."
720 MR=INT(ALTIMETER/30):IF MR>20 THEN MR=20
730 PRINT ":";LEFT$(Q$,MR);">"
740 PRINT ":FUEL     :"INT(FUEL)
750 PRINT ":";LEFT$(Q$,20-INT(FUEL/750));">"
760 PRINT ":-------------------------------:"
770 PRINT ":ELEVATION:"ELEVATE": ";:GOSUB 2210:PRINT U$
780 IF UFLAG=1 THEN PRINT ":";TAB(5);"> UNDERCARRIAGE DOWN <
 :"
790 IF UFLAG=0 THEN PRINT ":";TAB(6);"> UNDERCARRIAGE UP <
 :"
800  ENDPROC
810 REM ******************************
820 REM ASSIGN HORIZON/COMPASS
830 IF ABS(INT(WA+.5))=3 THEN GOSUB 980
840 IF ABS(INT(WA+.5))=2 THEN GOSUB 1070
850 IF ABS(INT(WA+.5))=1 THEN GOSUB 1160
860 IF INT(WA+.5)=0 THEN GOSUB 1250
870 REM NEXT TWO LINES USED TO GRADUALLY STRAIGHTEN UP WINGS
880 IF WA>0 THEN WA=WA-.2
890 IF WA<0 THEN WA=WA+.2
900 IF WA>.2 THEN 1350
910 FOR Z=1 TO 7
920 M$(8-Z)=A$(Z)
930 NEXT Z
940 FOR Z=1 TO 7
950 A$(Z)=M$(Z)
960 NEXT Z
970 GOTO 1350
980 REM WA=3 OR -3
990 A$(1)="              * "
1000 A$(2)="            **  "
1010 A$(3)="          **    "
1020 A$(4)="        **      "
1030 A$(5)="      **        "
1040 A$(6)="    **          "
1050 A$(7)="**              "
1060 RETURN
```

```
1070 REM WA=2 OR -2
1080 A$[1]="                  "
1090 A$[2]="              **  "
1100 A$[3]="          ***     "
1110 A$[4]="      ***         "
1120 A$[5]="  ***             "
1130 A$[6]="***               "
1140 A$[7]="                  "
1150 RETURN
1160 REM WA=1 OR -1
1170 A$[1]="                  "
1180 A$[2]="                  "
1190 A$[3]="            *****"
1200 A$[4]="    ******        "
1210 A$[5]="****              "
1220 A$[6]="                  "
1230 A$[7]="                  "
1240 RETURN
1250 REM WA=0
1260 A$[1]="                  "
1270 A$[2]="                  "
1280 A$[3]="                  "
1290 A$[4]="****************"
1300 A$[5]="                  "
1310 A$[6]="                  "
1320 A$[7]="                  "
1330 RETURN
1340 REM ****************************
1350 REM ASSIGN COMPASS STRINGS
1360 F2=ANG-F1
1370 IF F2<0 THEN FA=INT[(F2+375)/30]
1380 IF F2>=0 THEN FA=INT[(F2+15)/30]
1390 IF FA=12 THEN FA=0
1400 C$[1]="   .N.  "
1410 IF FA=11 THEN C$[2]="  .@:.. ":GOTO 1450
1420 IF FA=0 THEN C$[2]="  ..@.. ":GOTO 1450
1430 IF FA=1 THEN C$[2]="  ..:@. ":GOTO 1450
1440 C$[2]="  ..:.. "
1450 IF FA=10 THEN C$[3]= " .@ : ..":GOTO 1480
1460 IF FA=2 THEN C$[3]= " .. : @.":GOTO 1480
1470 C$[3]=" .. : .."
```

```
1480 IF FA=9 THEN C$(4)=" W@-X--E":GOTO 1510
1490 IF FA=3 THEN C$(4)=" W--X-@E":GOTO 1510
1500 C$(4)=" W--X--E"
1510 IF FA=8 THEN C$(5)=" .@ : ..":GOTO 1540
1520 IF FA=4 THEN C$(5)=" .. : @.":GOTO 1540
1530 C$(5)=" .. : .."
1540 IF FA=7 THEN C$(6)="  .@:.. ":GOTO 1580
1550 IF FA=6 THEN C$(6)="  ..@.. ":GOTO 1580
1560 IF FA=5 THEN C$(6)="  ..:@. ":GOTO 1580
1570 C$(6)="  ..:.. "
1580 C$(7)="   .S.  "
1590 IF ANG>360 THEN ANG=ANG-360
1600 F2=ANG
1610 IF W>0 THEN W=W-.4
1620 IF W<0 THEN W=W+.4
1630 RETURN
1640 REM STALL/FALL
1650 IF STALL=-1 THEN 1710
1660 FOR J=1 TO 10
1670 PRINT TAB(J);"YOU HAVE STALLED!"
1680 NEXT J
1690 AIRSPEED=AIRSPEED/4
1700 RETURN
1710 FOR J=1 TO 10
1720 PRINT TAB(J);"UNCONTROLLED DIVE!!"
1730 PRINT TAB(21-J);"PULL UP!!"
1740 NEXT J
1750 ALTIMETER=4*ALTIMETER/5
1760 RETURN
1770 REM ****************************
1780 REM CRASH
1790 CRASH=1
1800 ALTIMETER=0
1810 M$="** *C R**  A ** S* H* !!*":REM 25 CHARACTERS LONG
1820 FOR J=1 TO 20
1830 PRINT TAB(J);"CRASH!"
1840 PRINT TAB(21-J);"CRASH!"
1850 NEXT J
1860 FOR J=1 TO 7
1870 G=INT(RND(1)*11)+1
1880 A$(J)=MID$(M$,G,14)
```

```
1890 NEXT J
1900 RETURN
1910 REM ****************************
1920 REM ADJUST HORIZON
1930 G$="                    ":REM 14 SPACES
1940 ON EV+3 GOSUB 1960,2020,2070,2080,2140
1950 RETURN
1960 REM EV=-2
1970 FOR J=1 TO 4
1980 A$(J)=A$(J+3)
1990 NEXT J
2000 A$(5)=G$:A$(6)=G$:A$(7)=G$
2010 RETURN
2020 REM EV=-1
2030 FOR J=1 TO 6
2040 A$(J)=A$(J+1)
2050 NEXT J
2060 A$(7)=G$
2070 RETURN:REM EV=0
2080 REM EV=1
2090 FOR J=6 TO 1 STEP -1
2100 A$(J+1)=A$(J)
2110 NEXT J
2120 A$(1)=G$
2130 RETURN
2140 REM EV=2
2150 FOR J=4 TO 1 STEP -1
2160 A$(J+3)=A$(J)
2170 NEXT J
2180 A$(1)=G$:A$(2)=G$:A$(3)=G$
2190 RETURN
2200 REM ****************************
2210 REM INPUT INTO COMMAND NAME
2220 U$="________"
2230 IF X$=" " THEN U$="THROTTLE ON"
2240 IF X$="." THEN U$="THROTTLE OFF"
2250 IF X$="Q" AND ALTIMETER>0 THEN U$="CLIMB"
2260 IF X$="Q" AND ALTIMETER=0 THEN U$="NOSE UP"
2270 IF X$="A" THEN U$="NOSE DOWN"
2280 IF X$="Z" THEN U$="BANK LEFT"
2290 IF X$="M" THEN U$="BANK RIGHT"
```

```
2300 RETURN
2310 REM ****************************
2320  DEFPROCinitial
2330 PRINT RND[TIME]
2340 MODE 7
2350 Q$="————————————————————————":REM 21 CHARACTERS IN STRING

2360 UFLAG=1:REM UNDERCARRIAGE - 1 - DOWN, 0 - UP
2370 EFLAG=0:REM CLIMB RATE
2380 ANG=0:TAKEOV=0:LAND=0
2390 AIRSPEED=0
2400 DIST=0:REM DISTANCE COVERED 'RANGE'
2410 ALTIMETER=0
2420 ELEVATE=0:REM ANGLE OF ELEVATION
2430 WA=0:REM 'WING ANGLE; USED IN HORIZON PRINTOUT
2440 FUEL=750:CRASH=0:F2=0:F1=0:REM FOR DIRECTION CHANGE/COM
PASS ROUTINE
2450 CLOCK=0:REM TIME
2460 X$=""
2470  ENDPROC
2480 REM ALL TAKE-OFFS ARE INTO THE NORTH [SPEED 45-60; ELEV
ATION >10]
2490 REM LANDING DIRECTION SHOWN BELOW COMPASS
2500 REM YOU MUST BE WITHIN 12 DEGREES OF THIS FOR A SUCCESS
FUL LANDING
```

Spectrum+/Spectrum Listings

Space Landing

```
10 REM SPACE LANDING SIMULATION
I
15 REM SPECTRUM VERSION
20 RANDOMIZE:INK 7:PAPER 1:BORDE
R 7
30 REM ************************
****
40 REM SET STARTING VALUES
50 FUEL=200+INT(RND*40)
60 VELOCITY=INT(RND*20)-6
70 HEIGHT=500-INT(RND*10)
80 CLS
90 PRINT " FUEL";TAB(12);" VELOC
ITY";TAB(24);" HEIGHT"
100 REM ************************
*****
110 REM MAJOR CYCLE
120 GOSUB 430
130 IF FUEL<=0 THEN LET FUEL=0:L
ET THRUST=0:GOTO 170
140 LET A$=INKEY$
150 IF A$<"0" OR A$>"9" THEN GOT
O 140
160 LET THRUST=VAL(A$)+.1
170 LET FUEL=FUEL-THRUST
180 LET FLAG=THRUST-2
190 LET THRUST=0
200 LET HEIGHT=HEIGHT+VELOCITY+F
LAG/4
210 LET VELOCITY=VELOCITY+FLAG
220 IF HEIGHT<=10 THEN GOTO 240
230 IF HEIGHT>10 THEN GOTO 120
240 IF VELOCITY>-9 AND VELOCITY<
5 THEN GOTO 290
250 GOSUB 410
260 CLS:PRINT FLASH 1;"      YOU
 HAVE CRASHED INTO      THE S
URFACE..."
270 IF HEIGHT>0 THEN LET HEIGHT=
-HEIGHT
280 GOTO 320
290 CLS:PRINT BRIGHT 1;"   YOU H
AVE LANDED SAFELY!   "
300 PRINT "YOUR SKILL RATING IS"
;INT(-1000*FUEL/(VELOCITY-HEIGHT
))
310 LET HEIGHT=0
320 GOSUB 410
330 PRINT "FINAL INSTRUMENT READ
INGS WERE:"
340 PRINT " FUEL";TAB(12);" VELO
CITY";TAB(24);" HEIGHT"
350 GOSUB 430
360 GOSUB 410
370 IF HEIGHT>=0 THEN STOP
380 IF HEIGHT<0 THEN PRINT "NEW
CRATER ON MOON";INT(ABS(100*(HEI
GHT+.2)/3))/100;"METRES DEEP!"
390 PRINT "YOUR SKILL RATING IS
"INT(100*FUEL/(VELOCITY-HEIGHT))

400 STOP
410 PRINT "-------------------
-----------"
420 RETURN
430 PRINT INT(100*FUEL)/100;
440 PRINT TAB(12);-INT(100*VELOC
ITY)/100;
450 IF HEIGHT>=0 THEN PRINT TAB(
24);INT(100*HEIGHT)/100
460 IF HEIGHT<0 THEN PRINT
470 RETURN
```

Monte Carlo

```
10 REM MONTE CARLO DEMONSTRATION
15 REM SPECTRUM VERSION
20 GOSUB 370:REM INITIALIZE
30 REM ******************
40 REM MAJOR CYCLE
50 GOSUB 100:REM PRINT
60 IF P=EP AND Q=EQ THEN PRINT:P
RINT "DEMONSTRATION OVER":BEEP .
4,10:STOP
70 GOSUB 230:REM GENERATE MOVE
80 GOTO 50
90 REM ******************
100 REM PRINTOUT
110 LET A$(P,Q)="O"
120 LET M=M+1
130 CLS:PRINT:PRINT
140 PRINT "MOVE: ";M
150 FOR X=1 TO 10
160 FOR Y=1 TO 10
170 PRINT A$(X,Y);" ";
180 NEXT Y
190 PRINT
200 NEXT X
210 RETURN
220 REM ******************
230 REM GENERATE MOVE
240 LET A$(P,Q)="."
250 LET G=0
260 LET T=INT(RND*4)+1
270 GOSUB 300+(T*10)
280 IF G=0 THEN GOTO 260
290 IF G=1 AND RND>.5 THEN GOTO
260
300 RETURN
310 IF P>1 THEN LET P=P-1:LET G=
G+1:RETURN
320 IF P<10 THEN LET P=P+1:LET G
=G+1:RETURN
330 IF Q>1 THEN LET Q=Q-1:LET G=
G+1:RETURN
340 IF Q<10 THEN LET Q=Q+1:LET G
=G+1:RETURN
350 RETURN
360 REM ******************
370 REM INITIALIZE
380 PAPER 1:INK 7:BORDER 7:CLS
390 RANDOMIZE
400 DIM A$(10,10)
410 FOR X=1 TO 10
420 FOR Y=1 TO 10
430 LET A$(X,Y)="."
440 NEXT Y
450 NEXT X
460 PRINT:PRINT:LET M=0
470 PRINT "ENTER FIRST START CO-
ORDINATE (LESS     THAN 10)"
480 INPUT P
490 IF P<1 OR P>10 THEN GOTO 480

500 PRINT "ENTER SECOND START CO
-ORDINATE (LESS     THAN 10)"
510 INPUT Q
520 IF Q<1 OR Q>10 THEN GOTO 510

530 PRINT:PRINT
540 PRINT "ENTER FIRST END CO-OR
DINATE (LESS       THAN 10)"
550 INPUT EP
560 IF EP=P OR EP<1 OR EP>10 THE
N GOTO 550
570 PRINT "ENTER SECOND END CO-O
RDINATE (LESS      THAN 10)"
580 INPUT EQ
590 IF EQ=Q OR EQ<1 OR EQ>10 THE
N GOTO 580
600 LET A$(P,Q)="O"
610 LET A$(EP,EQ)="X"
620 RETURN
```

Simultaneous Equations

```
10 REM SIMULTANEOUS EQUATIONS
15 REM SPECTRUM VERSION
20 CLS
30 RANDOMIZE: INK 7: PAPER 1: BO
RDER 7
40 LET HS=0
50 LET FD=RND
60 PRINT:PRINT "DECAY FACTOR IS
";FD
70 GOSUB 550
80 CLS
90 PRINT:PRINT
100 PRINT "ENTER NUMBER OF CELL
X TO                    START (LESS T
HAN 40)"          110 INPUT CP:IF
CP<1 OR CP>39 THEN GOTO 110
110 INPUT CP:IF CP<1 OR CP>39 TH
EN 110
120 PRINT:PRINT
130 PRINT "WE HAVE ";CP;" X CELL
S"
140 PRINT:PRINT
150 PRINT "ENTER NUMBER OF CELL
Y TO                    START (LESS T
HAN 40)"
160 INPUT EP:IF EP<1 OR EP>39 TH
EN GOTO 160
170 CLS:PRINT:PRINT "PLEASE STAN
D BY...":BEEP .07,10:BEEP .07,15
180 GOSUB 550:CLS
190 LET DA=1
200 IF CP>EP/FD THEN LET CP=EP/F
D
210 PRINT "____________________"
220 PRINT "TIME ELAPSED: ";DA
230 PRINT INT(CP);" CELL X"
240 PRINT INT(EP);" CELL Y"
250 REM ************************
***
260 REM MAJOR CYCLE
270 GOSUB 550
280 LET DA=DA+1
290 PRINT "____________________"
300 PRINT "TIME ELAPSED: ";DA
310 IF CP>EP/FD THEN LET CP=EP/F
D
320 REM EQUATIONS FOLLOW; MODIFY
 PARTS          OF THEM TO SEE WH
AT HAPPENS
330 LET CP=CP+((8*CP-CP*EP/3)*FD
)
340 LET EP=EP+((4*EP-EP*CP)*.01)

350 PRINT INT(CP);" CELL X"
360 PRINT INT(EP);" CELL Y"
370 IF EP<2 OR CP<2 THEN GOTO 41
0
380 GOSUB 550
390 GOTO 280
400 REM ************************
***
410 IF DA>HS THEN LET HS=DA
420 PRINT:PRINT
430 PRINT "YOUR CELL CLASH SIMUL
ATION          SURVIVED"
440 PRINT "FOR ";DA;" TIME PERIO
DS."
450 PRINT "____________________
____________"
460 PRINT "THE BEST SURVIVAL TIM
E SO FAR IS ";HS
470 GOSUB 550
480 PRINT "____________________
____________"
490 PRINT "DO YOU WANT A NEW RUN
 (Y OR N)?"
500 LET A$=INKEY$
510 IF A$<>"Y" AND A$<>"N" THEN
```

```
GOTO 500
520 IF A$="Y" THEN CLS:GOTO 60
530 PRINT "OK":PRINT:PRINT:STOP
540 REM ************************
***
550 FOR J=1 TO 800:NEXT J
560 RETURN
```

Robot Logo

```
0 REM
100 REM ROBOT LOGO
105 REM SPECTRUM VERSION
110 GOSUB 1730:REM INITIALIZE
120 GOTO 490
125 REM NOTE THAT COMMANDS SHOUL
D BE ENTERED AS DATA STATEMENTS
AND ENCLOSED WITHIN SPEECH MARKS

130 REM **************************
****
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM **************************
****
340 REM INT UX,AX
350 LET UX=INT(UX+.5):LET AX=INT
(AX+.5)
360 RETURN
370 REM **************************
****
380 REM          PRINT OUT
390 CLS
400 PRINT "STEP"PSN" > ";A$(PSN)
:PRINT
410 FOR J=1 TO DEPTH
420 FOR K=1 TO BREADTH
430 PRINT Z$(J,K);
440 NEXT K
450 PRINT
460 NEXT J
470 RETURN
480 REM **************************
****
490 REM READ PROGRAM
495 LET N$="":RESTORE:LET COUNT=
0
500 LET COUNT=COUNT+1
510 READ A$(COUNT)
520 IF A$(COUNT,1)="*" THEN GOTO
 550
530 IF COUNT<20 THEN GOTO 500
540 REM **************************
****
550 REM EXECUTE PROGRAM
560 LET PSN=0:REM PROGRAM STEP N
UMBER
570 LET PSN=PSN+1
580 IF PSN=21 THEN GOTO 580:REM
END
600 LET M$=A$(PSN, TO 12)
610 IF M$(1)="*" THEN BEEP .01,I
NT(RND*60)-25:GOTO 610:REM END
620 LET N$=M$( TO 2)
630 IF N$="ST" THEN GOTO 560:REM
 START AGAIN
640 IF N$="PR" THEN GOSUB 380:RE
M PRINTOUT
650 IF N$="FO" THEN GOSUB 780
660 IF N$="BA" THEN GOSUB 940
670 IF N$="TU" THEN GOSUB 1000
```

```
680 IF N$="HO" THEN GOSUB 1160
690 IF N$="CL" THEN GOSUB 1220
700 IF N$="GO" THEN GOSUB 1260
710 IF N$="RA" THEN GOSUB 1400
720 IF N$="RE" THEN GOSUB 1460
730 IF N$="EN" THEN GOSUB 1530
740 IF N$="FA" THEN GOSUB 1580
760 GOTO 570
770 REM ************************
****
780 REM              FORWARD
790 LET M$=M$[4 TO ]
800 IF CODE[M$]=87 THEN LET M$=M
$[6 TO ]
810 LET F$="F"
820 LET NUM=VAL[M$]
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN GOT
O 880
850 IF AX<1 OR AX>BREADTH THEN G
OTO 880
860 LET Z$[UX,AX]=T$
880 IF F$="F" THEN LET UX=UX+UP:
LET AX=AX+AC*2
890 IF F$="B" THEN LET UX=UX-UP:
LET AX=AX-AC*2
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ************************
****
940 REM              BACK
950 LET M$=M$[4 TO ]
960 IF CODE[M$]=75 THEN LET M$=M
$[3 TO ]
970 LET F$="B"
980 GOTO 820
990 REM ************************
****
1000 REM             TURN
1010 LET M$=M$[4 TO ]
1020 IF CODE[M$]=78 THEN LET M$=
M$[3 TO]
1030 LET NUM=VAL[M$]
1040 LET Y=INT[[NUM+17.5]/45]
1050 IF Y=0 OR Y=8 THEN RETURN
1060 FOR J=1 TO Y
1070 IF UP=-1 AND AC=0 THEN LET
AC=1:GOTO 1130
1080 IF UP=0 AND AC=1 THEN LET U
P=1:GOTO 1130
1090 IF UP=1 AND AC=0 THEN LET A
C=-1:GOTO 1130
1100 IF UP=0 AND AC=-1 THEN LET
UP=-1:GOTO 1130
1110 IF UP=-1 AND AC=-1 OR UP=1
AND AC=1 THEN LET AC=0:GOTO 1130

1120 IF UP=-1 AND AC=1 OR UP=1 A
ND AC=-1 THEN LET UP=0
1130 NEXT J
1140 RETURN
1150 REM ************************
*****
1160 REM              HOME
1170 LET AX=INT[[BREADTH+.5]/2]
1180 LET UX=INT[[DEPTH+.5]/2]
1190 LET UP=-1:LET AC=0:REM FACE
S UP
1200 RETURN
1210 REM ************************
*****
1220 REM              CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ************************
*****
1260 REM              GO X,Y
1270 LET P=0
1280 LET P=P+1
1290 IF M$[P]="," THEN GOTO 1320
```

```
1300 IF P<LEN(M$) THEN GOTO 1280
1310 RETURN:REM ERROR
1320 LET UX=VAL(M$(4 TO P-1))
1330 LET AX=VAL(M$(P+1 TO ))
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN GO
TO 1380
1360 IF AC<1 OR AC>BREADTH THEN
GOTO 1380
1370 Z$(UX,AX)=R$
1380 RETURN
1390 REM ************************
*****
1400 REM                RANDOM
1410 LET AX=INT(RND*BREADTH):IF
AX<1 THEN LET AX=1
1420 LET UX=INT(RND*DEPTH):IF UX
<1 THEN LET UX=1
1430 LET Z$(UX,AX)=R$
1440 RETURN
1450 REM ************************
*****
1460 REM                REPEAT
1470 LET M$=M$(4 TO)
1480 IF CODE(M$)=69 THEN LET M$=
M$(5 TO)
1490 LET RECOUNT=VAL(M$)
1500 LET MARKER=PSN
1510 RETURN
1520 REM ************************
*****
1530 REM              END REPEAT
1540 LET RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN LET PSN=M
ARKER
1560 RETURN
1570 REM ************************
*****
1580 REM                FACE
1590 LET M$=M$(4 TO )
1600 IF CODE(M$)=69 THEN LET M$=
M$(3 TO )
1610 LET NUM=VAL(M$)
1620 LET Y=INT((NUM+17.5)/45)*45
1630 IF Y=0 OR Y=360 THEN LET UP
=-1:LET AC=0
1640 IF Y=45 THEN LET UP=-1:LET
AC=1
1650 IF Y=90 THEN LET UP=0:LET A
C=1
1660 IF Y=135 THEN LET UP=1:LET
AC=1
1670 IF Y=180 THEN LET UP=1:LET
AC=0
1680 IF Y=225 THEN LET UP=1:LET
AC=-1
1690 IF Y=270 THEN LET UP=0:LET
AC=-1
1700 IF Y=315 THEN LET UP=-1:LET
 AC=-1
1710 RETURN
1720 REM ************************
*****
1730 REM INITIALIZE
1740 PAPER 1:INK 7:BORDER 7:CLS
1750 RANDOMIZE
1770 LET BREADTH=32:REM CHARACTE
RS ACROSS
1780 LET DEPTH=22:REM CHARACTERS
 DOWN
1790 LET BREADTH=BREADTH-1
1800 LET DEPTH=DEPTH-3
1810 LET UP=-1:LET AC=0:REM STAR
TS FACING UP
1820 DIM A$(20,12):REM FOR ROBOT
 PROGRAM
1830 DIM Z$(DEPTH,BREADTH):REM D
ISPLAY
1840 LET T$="X":LET R$="X":REM P
UT SYMBOL HERE YOU          WANT
```

```
TO USE FOR ROBOT'S TRAIL
1850 LET AX=0:LET UX=0
1860 REM FILL ARRAY WITH SPACES
1870 FOR J=1 TO DEPTH
1880 FOR K=1 TO BREADTH
1890 LET Z$(J,K)=" "
1900 NEXT K
1910 NEXT J
1915 CLS
1920 RETURN
```

Point Duty Robot

```
100 REM POINT-DUTY ROBOT
105 REM SPECTRUM VERSION
110 GOSUB 1730:REM INITIALIZE
120 GOTO 490
125 REM NOTE THAT COMMANDS SHOUL
D  BE ENTERED AS DATA STATEMENTS
 AND ENCLOSED WITHIN SPEECH MARK
S
130 REM ************************
****
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 DATA "*"
330 REM ************************
****
340 REM INT UX,AX
350 LET UX=INT(UX+.5):LET AX=INT
(AX+.5)
360 RETURN
480 REM ************************
****
490 REM READ PROGRAM
495 LET N$="":RESTORE:LET COUNT=
0
500 LET COUNT=COUNT+1
510 READ A$(COUNT)
520 IF A$(COUNT)="*" THEN GOTO 5
50
530 IF COUNT<20 THEN GOTO 500
540 REM ************************
****
550 REM EXECUTE PROGRAM
560 LET PSN=0:REM PROGRAM STEP N
UMBER
570 LET PSN=PSN+1
580 IF PSN=21 THEN GOTO 580:REM
END
600 LET M$=A$(PSN, TO 12)
610 IF M$(1)="*" THEN BEEP .01,I
NT (RND*60)-25:GOTO 610:REM END
620 LET N$=M$( TO 2)
630 IF N$="ST" THEN GOTO 560:REM
 START AGAIN
650 IF N$="FO" THEN GOSUB 780
660 IF N$="BA" THEN GOSUB 940
670 IF N$="TU" THEN GOSUB 1000
680 IF N$="HO" THEN GOSUB 1160
700 IF N$="GO" THEN GOSUB 1260
710 IF N$="RA" THEN GOSUB 1400
720 IF N$="RE" THEN GOSUB 1460
730 IF N$="EN" THEN GOSUB 1530
740 IF N$="FA" THEN GOSUB 1580
760 GOTO 570
770 REM ************************
****
780 REM              FORWARD
790 LET M$=M$( TO 4)
800 IF CODE(M$)=87 THEN LET M$=M
$(6 TO )
810 LET F$="F"
820 LET NUM=VAL(M$)
830 FOR E=1 TO NUM
840 IF UX<1 OR UX>DEPTH THEN GOT
```

```
0 880
850 IF AX<1 OR AX>BREADTH THEN G
OTO 880
860 PLOT AX,UX
880 IF F$="F" THEN LET UX=UX-DV:
LET AX=AX+HB
890 IF F$="B" THEN LET UX=UX+UP:
LET AX=AX-AC
900 GOSUB 340
910 NEXT E
920 RETURN
930 REM ***********************
****
940 REM              BACK
950 LET M$=M$[4 TO ]
960 IF CODE[M$]=75 THEN LET M$=M
ID$[3 TO ]
970 LET F$="B"
980 GOTO 820
990 REM ***********************
****
1000 REM              TURN
1010 LET M$=M$[4 TO ]
1020 IF CODE[M$]=78 THEN LET M$=
MID$[3 TO ]
1030 NUM=VAL[M$]
1040 LET Y=INT[(NUM+11.25)/22.5]

1050 IF Y=0 OR Y=16 THEN RETURN
1060 FOR J=1 TO Y
1065 IF UP=-2 AND AC=0 OR UP=2 A
ND AC=2 THEN LET AC=1:GOTO 1130
1070 IF UP=-2 AND AC=1 THEN LET
AC=2:GOTO 1130
1075 IF (UP=-2 AND AC=2) OR (UP=
0 AND AC=-2) THEN LET UP=-1:GOTO
 1130
1080 IF (UP=-1 AND AC=2) OR (UP=
1 AND AC=-2) THEN LET UP=0:GOTO
1130
1085 IF (UP=0 AND AC=2) OR (UP=2
 AND AC=-2) THEN LET UP=1:GOTO 1
130
1090 IF UP=1 AND AC=2 THEN LET U
P=2:GOTO 1130
1095 IF UP=2 AND AC=1 THEN LET A
C=0:GOTO 1130
1100 IF UP=2 AND AC=0 THEN LET A
C=-1:GOTO 1130
1105 IF UP=2 AND AC=-1 THEN LET
AC=-2:GOTO 1130
1110 IF UP=-1 AND AC=-2 THEN LET
 UP=-2:GOTO 1130
1115 IF UP=-2 AND AC=-2 THEN LET
 AC=-1:GOTO 1130
1120 IF UP=-2 AND AC=-1 THEN LET
 AC=0
1130 NEXT J
1135 LET HB=AC:LET DV=UP
1140 RETURN
1150 REM ***********************
*****
1160 REM              HOME
1170 LET AX=INT[(BREADTH+.5)/2]
1180 LET UX=INT[(DEPTH+.5)/2]
1190 LET UP=-2:LET AC=0:REM FACE
S UP
1200 RETURN
1210 REM ***********************
*****
1220 REM              CLEAN
1230 GOSUB 1870
1240 RETURN
1250 REM ***********************
*****
1260 REM              GO X,Y
1270 LET P=0
1280 LET P=P+1
1290 IF M$(P)=="," THEN GOTO 132
0
1300 IF P<LEN[M$] THEN GOTO 1280
```

```
1310 RETURN:REM ERROR
1320 LET UX=VAL[M$[4 TO P-1]]
1330 LET AX=VAL[M$[P+1 TO ]]
1340 GOSUB 340
1350 IF UX<1 OR UX>DEPTH THEN GO
TO 1380
1360 IF AC<1 OR AC>BREADTH THEN
GOTO 1380
1370 PLOT AX,UX
1380 RETURN
1390 REM ************************
*****
1400 REM              RANDOM
1410 LET AX=INT[RND*BREADTH]:IF
AX<1 THEN LET AX=1
1420 LET UX=INT[RND*DEPTH]:IF UX
<1 THEN LET UX=1
1430 PLOT AX,UX
1440 RETURN
1450 REM ************************
*****
1460 REM              REPEAT
1470 LET M$=MID$[4 TO ]
1480 IF CODE[M$]=69 THEN LET M$=
MID$[5 TO ]
1490 LET RECOUNT=VAL[M$]
1500 LET MARKER=PSN
1510 RETURN
1520 REM ************************
*****
1530 REM          END REPEAT
1540 LET RECOUNT=RECOUNT-1
1550 IF RECOUNT>0 THEN LET PSN=M
ARKER
1560 RETURN
1570 REM ************************
*****
1580 REM              FACE
1590 LET M$=MID$[4 TO ]
1600 IF CODE[M$]=69 THEN LET M$=
M$[3 TO ]
1610 LET NUM=VAL[M$]
1620 LET Y=INT[[NUM+11.25]/22.5]
*22.5
1630 IF Y=0 OR Y=360 THEN LET UP
=-2:AC=0
1635 IF Y=22.5 THEN LET UP=-2:AC
=1
1640 IF Y=45 THEN LET UP=-2:AC=2
1645 IF Y=67.5 THEN LET UP=-1:AC
=2
1650 IF Y=90 THEN LET UP=0:AC=2
1655 IF Y=112.5 THEN LET UP=1:AC
=2
1660 IF Y=135 THEN LET UP=2:AC=2
1665 IF Y=157.5 THEN LET UP=2:AC
=1
1670 IF Y=180 THEN LET UP=2:AC=0
1675 IF Y=202.5 THEN LET UP=2:AC
=-1
1680 IF Y=225 THEN LET UP=2:AC=-
2
1685 IF Y=247.5 THEN LET UP=1:AC
=-2
1690 IF Y=270 THEN LET UP=0:AC=-
2
1695 IF Y=292.5 THEN UP=-1:AC=-2
1700 IF Y=315 THEN LET UP=-2:AC=
-2
1705 IF Y=337.5 THEN LET UP=-2:A
C=-1
1710 RETURN
1720 REM ************************
*****
1730 REM INITIALIZE
1735 LET HB=0:LET DV=0
1740 PAPER 1:INK 7:BORDER 7:CLS
1750 RANDOMIZE
```

```
1770 LET BREADTH=255:REM CHARACT
ERS ACROSS
1780 LET DEPTH=170:REM CHARACTER
S DOWN
1790 LET BREADTH=BREADTH-1
1800 LET DEPTH=DEPTH-3
1810 LET UP=-2:LET AC=0:REM STAR
TS FACING UP
1820 DIM A$(20,12):REM FOR ROBOT
 PROGRAM
1910 CLS
1920 RETURN
```

Connect Four

```
10 REM CONNECT FOUR
15 REM SPECTRUM VERSION
20 REM A. W. PEARSON
30 PAPER 7:  INK 0:  BORDER 7:
CLS
40 PRINT
50 PRINT
60 PRINT "CONNECT FOUR"
70 PRINT
80 PRINT "ENTER YOUR MOVE AS A N
UMBER"
90 PRINT "BETWEEN 1 AND 8, 0 FOR
 NEW GAME"
100 FOR F=1 TO 500:NEXT F
110 DIM A$(10,10):DIM B(10,2)
120 LET FLAG=0
130 REM CHANGE NEXT LINE FOR YOU
R OWN SYMBOLS IF YOU WISH
140 C$="M":H$="o":REM M FOR MACH
INE!
150 FOR F=1 TO 8
160 LET B(F,1)=6
170 NEXT F
180 FOR F=1 TO 6
190 FOR G=1 TO 8
200 LET A$(F,G)="."
210 NEXT G
220 NEXT F
230 REM ************************
*
240 REM ACCEPT HUMAN MOVE
250 GOSUB 430
260 PRINT:PRINT:PRINT "YOUR MOVE
..."
270 INPUT A
280 IF A=0 THEN RUN
290 IF A<1 OR A>8 THEN GOTO 270
300 LET L=0:LET A=A+1
310 IF A$(L+1,A)<>"." OR L=6 THE
N GOTO 340
320 LET L=L+1
330 GOTO 310
340 IF L=0 THEN GOTO 270
350 LET A$(L,A)=H$
360 LET B(A,1)=B(A,1)-1
370 GOSUB 430
380 GOSUB 560
390 GOSUB 430
400 GOTO 260
410 REM ************************
*
420 REM PRINT BOARD
425 BEEP .07,15
430 CLS
435 PRINT:PRINT:PRINT TAB(10);
440 FOR F=1 TO 6
450 FOR G=1 TO 8
460 PRINT A$(F,G);
470 NEXT G
480 PRINT TAB(10);
490 NEXT F
500 PRINT:PRINT INK 0:TAB (10);
"12345678"
510 PRINT
520 IF FLAG=1 THEN PRINT "I HAVE
 WON":BEEP 1,-15:STOP
530 RETURN
540 REM ************************
*
550 REM SPECTRUM MOVES
560 PRINT "MY MOVE..."
570 LET MV=0
580 FOR F=1 TO 8
590 LET B(F,2)=0
600 NEXT F
610 FOR F=1 TO 8
620 FOR X=-1 TO 1
630 FOR Y=-1 TO 1
```

```
640 IF B[F,1]=0 THEN GOTO 680
650 IF A$[B[F,1]+X,F+Y]="" OR A$
[B[F,1]+X,F+Y]="." THEN GOTO 680
660 IF A$[B[F,1]+X,F+Y]=H$ THEN
GOSUB 810
670 IF A$[B[F,1]+X,F+Y]=C$ THEN
GOSUB 910
680 NEXT Y
690 NEXT X
700 NEXT F
710 LET P=0
720 FOR F=1 TO 8
730 IF B[F,2]>P THEN LET P=B[F,2
]:LET N=F
740 NEXT F
750 LET A$[B[N,1],N]=C$
760 LET B[N,1]=B[N,1]-1
770 LET N=0
780 LET P=0
790 RETURN
800 REM ************************
*
810 LET MV=2
820 LET M1=MV
830 IF A$[B[F,1]+[X*2],F+[Y*2]]=
H$ THEN LET MV=MV+10
840 IF A$[B[F,1]-X,F-Y]=H$ THEN
LET MV=MV+20
850 IF MV<>M1+10 THEN GOTO 870
860 IF A$[B[F,1]+[X*3],F+[Y*3]]=
H$ THEN LET MV=MV+1000
870 LET B[F,2]=B[F,2]+MV
880 LET M1=0
890 RETURN
900 REM ************************
*
910 LET MV=2
920 LET M1=MV
930 IF A$[B[F,1]+[X*2],F+[Y*2]]=
C$ THEN LET MV=MV+9
940 IF A$[B[F,1]-X,F-Y]=C$ THEN
LET MV=MV+20
950 IF MV<>M1+9 THEN GOTO 970
960 IF A$[B[F,1]+[X*3],F+[Y*3]]=
C$ THEN LET MV=MV+2000:LET FLAG=
1
970 LET B[F,2]=B[F,2]+MV
980 RETURN
```

Washington D.C.

```
10 REM WASHINGTON D.C.
15 REM SPECTRUM VERSION
20 GOSUB 1160:REM INITIALIZE
30 REM ****************************
**
40 REM MAJOR CYCLE
50 LET P=INT(P+(P*273/ML))
60 GOSUB 160:REM PRINTOUT
70 GOSUB 510:REM CALCULATE
80 REM NOW CHECK END GAME
90 GOSUB 710:REM STANDARD OF LIV
ING
100 GOSUB 780:REM INFLATION RATE

110 GOSUB 840:REM UNEMPLOYMENT
120 IF GAME=1 THEN CLS:GOTO 890
130 GOTO 50
140 REM ****************************
***
150 REM PRINTOUT
160 CLS
170 PRINT "PRESIDENT ";A$;":"
180 PRINT "YOUR ADMINISTRATION H
AS BEEN IN POWER FOR ";Y+Z/4;" Y
EARS"
190 PRINT "-------------------------
-------------"
200 PRINT "-----STATE OF THE NAT
ION----------"
210 PRINT "-------------------------
-------------"
220 PRINT "POPULATION ";P
230 PRINT "NO. UNEMPLOYED ";INT(
U);"   ";INT(100*U/P);"%"
240 PRINT "CURRENT WAGES $";WO;"
 INFLATION ";INT(IP);"%"
250 PRINT "GOVT. EXPENDITURE LAS
T QTR. $M";GE
260 PRINT "UNEMPLOYMENT COST $M;
"INT(10*GU)/10
270 PRINT "INCOME FROM TAXES $M;
"INT(GI*10)/10
280 PRINT "BUDGET SURPLUS(+)/DEF
ICIT(-) $M";INT(BD*10)/10
290 PRINT "GROSS DOMESTIC PRODUC
T $M";INT(GDP*10)/10
300 IF Y+Z/4>.5 THEN PRINT "CHAN
GE IN LIVING STANDARD ";INT((2*(
(RGDP/AGDP)*100)-100)/3);"%"
310 PRINT "-------------------------
-------------"
320 PRINT "PUB INVEST. ";Y;" QTR
 ";Z;"  $M";INT(IV*10)/10
330 PRINT "-------------------------
-------------"
340 PRINT "OK, PRESIDENT ";A$;".
..":BEEP .2,8
350 INPUT "ENTER GOVERNMENT SPEN
DING $M";GE
360 INPUT "ENTER COST OF WAGES $
M";WN
370 PRINT "IS YOUR ADMINISTRATIO
N IN FAVOUR"
380 PRINT "OF IMMIGRATION (Y/N)?
"
390 LET X$=INKEY$
400 IF X$<>"Y" AND X$<>"N" THEN
GOTO 390
410 PRINT TAB(16);"OK...";X$
420 BEEP .1,10: BEEP .1,7: BEEP
.3,13
430 IF X$<>"Y" THEN RETURN
440 PRINT "HOW MANY WILL YOU ALL
OW INTO THE US?"
450 INPUT M
460 IF M<0 THEN GOTO 450
470 LET P=P+M
480 RETURN
```

```
490 REM ************************** 780 REM CHECK INFLATION RATE
***                                790 IF IP<15 THEN RETURN
500 REM CALCULATIONS               800 LET GAME=1
510 LET CN=CN+(CN*IP/100)          810 FLAG=3
520 LET U=P*(GE+IV)/(CN*10)+P*(I   820 RETURN
P/1000)                            830 REM **************************
530 LET GU=U*WN/ML:REM UNEMPLOYM   ***
ENT COST                           840 REM CHECK UNEMPLOYMENT
540 LET GI=((P-U)*WN*.4)/ML):RE    850 IF INT(U*100/P)<15 THEN RETU
M INCOME FROM TAXE                 RN
550 LET BD=BD+GI-GU-GE:REM BUDGE   860 LET GAME=1
T DEFICIT                          870 LET FLAG=4
560 LET AGDP=AGDP*(1+(IP/100))     880 RETURN
570 LET GDP=GE+IV+((P-U)*WN/ML)    890 REM **************************
580 LET RGDP=GDP*440/AGDP          ***
590 LET IP=((GE+IV)/CN*.1+(WN/WO   900 REM END OF THE GAME
)/100)*100                         910 PRINT "PRESIDENT ";A$;", YOU
600 LET IV=(CN*67)/(IP*IP)         R"
610 LET WO=WN                      920 PRINT "ADMINISTRATION'S POOR
620 LET Z=Z+1:IF Z>4 THEN LET Z=    ECONOMIC"
1:LET Y=Y+1                        930 PRINT "PERFORMANCE HAS LED T
630 RETURN                         O AN UNACCEPTABLE"
640 REM ************************** 940 IF FLAG=1 THEN PRINT "BUDGET
***                                 DEFICIT"
650 REM CHECK BUDGET DEFICIT       950 IF FLAG=2 THEN PRINT "DROP I
660 IF BD>-1000 THEN RETURN        N THE STANDARD OF LIVING"
670 LET GAME=1                     960 IF FLAG=3 THEN PRINT "RISE I
680 LET FLAG=1                     N THE INFLATION RATE"
690 RETURN                         970 IF FLAG=4 THEN PRINT "RISE I
700 REM ************************** N UNEMPLOYMENT"
***                                980 PRINT "          AMONG OTHER
710 REM CHECK STANDARD OF LIVING   THINGS..."
                                   990 PRINT "_____________________
720 IF Y<.75 THEN RETURN           _____"
730 IF INT((2*((RGDP/AGDP)*100)-   1000 PRINT "THE LACK OF CONFIDEN
100)/3)>-15 THEN RETURN            CE IN YOUR"
740 LET GAME=1                     1010 PRINT "ADMINISTRATION IS SO
750 LET FLAG=2                      BAD THERE ARE"
760 RETURN                         1020 PRINT "CALLS FOR YOU TO RES
770 REM ************************** IGN...YOU STEP"
***                                1030 PRINT "ASIDE TO ALLOW THE V
```

```
ICE-PRESIDENT TO"
1040 PRINT "        OCCUPY THE OV
AL OFFICE"
1050 FOR H=1 TO 1000:NEXT H
1060 PRINT "____________________
_____________"
1070 PRINT "YOU WERE PRESIDENT F
OR ";Y+(Z*.25);" YEARS"
1080 PRINT "DURING YOUR TERM OF
OFFICE, THE"
1090 PRINT "POPULATION ROSE BY "
;P-3*ML
1100 PRINT "THE UNEMPLOYMENT RAT
E BECAME ":PRINT INT(U*1000/P)/1
0;"%"
1110 PRINT "AND THE INFLATION RA
TE BECAME ":PRINT INT(10*IP)/10;
"%"
1120 PRINT "STANDARD OF LIVING C
HANGED BY":PRINT INT((2*((RGDP/A
GDP)*100)-100)/3);"%"
1130 PRINT "THE BUDGET SURPLUS/D
EFICIT WAS $M";INT(10*BD)/10
1140 END
1150 REM ************************
****
1160 REM INITIALIZATION
1170 PAPER 1: INK 7: BORDER 7: C
LS
1180 RANDOMIZE VAL(RIGHT$(TIME$,
2))
1190 LET ML=1000*1000
1200 LET P=3*ML:REM POPULATION
1210 LET U=P/10:REM UNEMPLOYMENT
1220 LET IV=236:REM INVESTMENT
1230 LET GE=118:REM GOVERNMENT E
XPENDITURE
1240 LET GU=0:REM COST OF UNEMPL
OYMENT
1250 LET GI=0:REM INCOME FROM TA
XES
1260 LET WN=100:REM NEW WAGES
1270 LET WO=100:REM OLD WAGES
1280 LET IP=5:REM INFLATION PERC
ENT
1290 LET GDP=440:REM GROSS DOMES
TIC PRODUCT
1300 LET AGDP=440:REM BASE YEAR
GDP
1310 LET RGDP=440:REM REAL GDP
1320 LET CN=354:REM ECONOMIC CON
STANT
1330 LET Z=1:LET GAME=0:LET FLAG
=0
1340 LET Y=0:REM YEAR
1350 PRINT "ENTER YOUR LAST NAME
"
1360 INPUT A$
1370 RETURN
```

Stock Market

```
10 REM STOCK MARKET
15 REM SPECTRUM VERSION
20 PAPER 1: INK 7: BORDER 7: CLS

30 RANDOMIZE
40 DIM S(5):DIM N(5):DIM P(5):DI
M D(5)
50 LET S(1)=1.49:LET S(2)=1.99:L
ET S(3)=2.49:LET S(4)=2.99:LET S
(5)=3.49
60 LET N(1)=2000:LET N(2)=1500:L
ET N(3)=1200:LET N(4)=1000:LET N
(5)=800
70 LET BB=265:LET TV=15000:LET Q
Q=15000:LET DAY=1
80 PRINT:PRINT "ENTER YOUR GOAL
FOR THIS SIMULATION,"
90 PRINT TAB(8);"$16,000 TO $100
,000"
100 INPUT GOAL
110 IF GOAL<16000 THEN PRINT "TO
O LOW!":GOTO 80
120 IF GOAL>100*1000 THEN PRINT
"TOO HIGH!":GOTO 80
130 REM **************************

140 REM MAJOR LOOP
145 BEEP  .15,INT(RND*30)-15
150 FOR C=1 TO 5
160 REM ADJUST THE 55 IN NEXT LI
NE TO   MODIFY GAME; 80 VERY HAR
D, 30 VERY EASY
170 LET D(C)=INT(RND*55)+1
180 LET P(C)=INT(RND*(100-D(C)))
+1
190 NEXT C
200 GOSUB 230
210 GOTO 460
220 REM **************************
230 REM PRINTOUT
240 CLS
250 PRINT "_____________________
_____________"
260 PRINT "DAY ";DAY;"   YOUR GO
AL IS $";GOAL
270 PRINT "_____________________
_____________"
280 PRINT "COMPANY NUMBER:"
290 PRINT TAB(1);1;TAB(8);2;TAB(
15);3;TAB(22);4;TAB(29);5
300 PRINT "CHANCE OF INCREASE (%
):"
310 PRINT TAB(1);P(1);TAB(8);P(2
);TAB(15);P(3);TAB(22);P(4);TAB(
29);P(5)
320 PRINT "CHANCE OF DECREASE (%
):"
330 PRINT TAB(1);D(1);TAB(8);D(2
);TAB(15);D(3);TAB(22);D(4);TAB(
29);D(5)
340 PRINT "CURRENT VALUE PER SHA
RE:"
350 PRINT "$";INT(S(1)*100)/100;
TAB(7);"$";INT(S(2)*100)/100;
360 PRINT TAB(14);"$";INT(S(3)*1
00)/100;TAB(21);"$";INT(S(4)*100
)/100;
370 PRINT TAB(28);"$";INT(S(5)*1
00)/100
380 PRINT "NO. OF SHARES HELD:"
390 PRINT TAB(1);N(1);TAB(8);N(2
);TAB(15);N(3);TAB(22);N(4);TAB(
28);N(5)
400 PRINT "BANK $";INT(BB);" TOT
AL WORTH $";TV
410 PRINT "_____________________
_____________"
420 IF TV>GOAL THEN PRINT "YOU'V
```

```
E HIT YOUR FINANCIAL GOAL!":STOP

430 RETURN
440 REM ****************************

450 REM          ** SELL **
460 PRINT "DO YOU WANT TO SELL A
NY SHARES [Y/N]?"
470 A$=INKEY$
480 IF A$<>"Y" AND A$<>"N" THEN
GOTO 470
490 IF A$="N" THEN GOTO 690
500 GOSUB 230
510 PRINT "WHICH ONES TO SELL?";

520 LET A$=INKEY$
530 IF A$<"1" OR A$>"5" THEN GOT
O 520
540 LET C=VAL[A$]
550 PRINT "    OK ";C
560 PRINT "HOW MANY OF ";C;" TO
SELL";
570 INPUT N
580 IF N>N[C] THEN PRINT "YOU DO
N'T HAVE THAT MANY!":GOTO 570
590 REM ****************************

600 REM ADJUST FIGURES AFTER SAL
E
610 LET BB=BB+S[C]*N:REM ADD VAL
UE TO BANK
620 LET N[C]=N[C]-N:REM SUBTRACT
 NO. SOLD
630 LET TV=0:REM SET TOTAL WORTH
 TO ZERO
640 REM NOW DETERMINE CURRENT WO
RTH
650 FOR C=1 TO 5
660 LET TV=TV+N[C]*S[C]
670 NEXT C
680 LET TV=TV+BB:REM ADD IN BANK
 BALANCE
690 GOSUB 230
700 REM ****************************

710 REM          ** BUY **
720 PRINT "DO YOU WANT TO BUY AN
Y SHARES [Y/N]?"
730 LET A$=INKEY$
740 IF A$<>"Y" AND A$<>"N" THEN
GOTO 730
750 IF A$="N" THEN GOTO 890
760 GOSUB 230
770 PRINT "WHICH COMPANY TO BUY?
";
780 LET A$=INKEY$
790 IF A$<"1" OR A$>"5" THEN GOT
O 780
800 LET C=VAL[A$]
810 PRINT "    OK ";C
820 PRINT "HOW MANY OF ";C;" TO
BUY";
830 INPUT N
840 IF N*S[C]>BB THEN PRINT "YOU
 DON'T HAVE ENOUGH MONEY!":GOTO
830
850 REM ****************************

860 REM ADJUST FIGURES AFTER BUY

870 LET BB=BB-S[C]*N
880 LET N[C]=N[C]+N
890 LET TV=0
900 FOR C=1 TO 5
910 LET TV=TV+N[C]*S[C]
920 NEXT C
930 LET TV=TV+BB
940 GOSUB 230
950 REM ****************************

960 REM  MODIFY ALL INDICATORS
970 LET TV=0
```

```
980 FOR C=1 TO 5
990 LET K=INT(RND*100)+1
1000 IF K<P(C) THEN LET S(C)=S(C
)*(1+(P(C)/1000))
1010 LET K=INT(RND*100)+1
1020 IF K<D(C) THEN LET S(C)=S(C
)/(1+(D(C)/1000))
1030 LET TV=TV+(S(C)*N(C))
1040 NEXT C
1050 LET TV=TV+BB
1060 LET QQ=QQ*1.005
1070 LET W=(TV*100/QQ)-100
1080 IF W=0 THEN LET W=.1
1090 LET W=W+6
1100 IF W<1 THEN LET W=1
1110 IF W>15 THEN LET W=15
1120 RESTORE
1130 FOR T=1 TO W
1140 READ A$
1150 NEXT T
1160 PRINT
1170 REM ************************
*
1180 REM GIVE RATING, START NEW
ROUND
1190 PRINT "YOUR RATING AFTER TH
AT ROUND OF"
1200 PRINT "TRADING IS '";A$;"'"

1210 PRINT:PRINT "   <PRESS SPAC
E TO CONTINUE >"
1220 IF INKEY$<>" " THEN GOTO 12
20:REM NOTE SPACE BETWEEN QUOTE
MARKS
1230 LET DAY=DAY+1
1240 GOTO 150
1250 DATA "HOPELESS","VERY, VERY
 POOR"
1260 DATA "TERRIBLE","AWFUL","BA
D"
1270 DATA "VERY ORDINARY","AVERA
GE"
1280 DATA "REASONABLE","A LITTLE
 ABOVE AVERAGE"
1290 DATA "FAIRLY GOOD","GOOD","
VERY GOOD"
1300 DATA "GREAT","EXCELLENT","S
UPERLATIVE"
```

Detroit City

```
10 REM DETROIT CITY
15 REM SPECTRUM VERSION
20 GOSUB 1640:REM INITIALIZE
30 GOTO 110
40 LET MT=MT+1:REM COUNTS MONTHS
50 GOSUB 650
60 IF TP>200 THEN GOTO 1560
70 PRINT "DO YOU WANT TO RESIGN
(Y/N)?"
80 GOSUB 1010
90 IF A$="Y" THEN PRINT "OK, CHI
EF":STOP
100 GOSUB 1380
110 GOSUB 650
120 FOR T=1 TO 1000:NEXT T
130 GOSUB 850
140 PRINT "DO YOU WANT TO EXPAND
 OUTPUT (Y/N)?"
150 GOSUB 1010
160 IF A$="Y" THEN GOTO 1080
170 IF SF=1 THEN GOTO 210
180 PRINT "DO YOU WANT TO SELL F
ACTORY 4 (Y/N)?"
190 GOSUB 1010
200 IF A$="Y" THEN GOTO 1250
210 GOSUB 650
220 INPUT "HOW MANY EMPLOYEES TO
 HIRE";HE
230 LET NE=NE+HE:IF HE>0 THEN GO
TO 260
240 INPUT "HOW MANY EMPLOYEES TO
 FIRE";HE
250 LET NE=NE-HE
260 GOSUB 650
270 LET P1=AS:REM SET P1 EQUAL T
O OLD PRICE
280 INPUT "WHAT IS YOUR SELLING
PRICE";AS
290 REM NEXT LINE REJECTS TOO BI
G A                    CHANGE IN SELL
ING PRICE
300 IF ABS(P1-AS)>2500 THEN PRIN
T "TOO BIG A CHANGE FOR THE MARK
ET":GOTO 280
310 CLS
320 PRINT:PRINT:PRINT
330 LET MI=INT(RND*4000)+48*1000
:REM THIS        MONTH'S SALES BY
 INDUSTRY
340 LET C=C+1:REM COUNTS NUMBER
OF MONTHS
350 IF C<3 THEN GOTO 470
360 LET M=INT(RND*10+1)/4:REM IN
FLATION
370 CLS
380 PRINT "INFLATION RATE THIS Q
UARTER IS ";M;"%"
390 PRINT "AVERAGE WAGES BILL WI
LL NOW RISE TO"
400 LET AW=(AW*M/100)+AW
410 PRINT TAB(8);"$";INT(AW);"
PER EMPLOYEE"
420 IF INKEY$<>"" THEN GOTO 420
430 PRINT:PRINT TAB(12);"ANY KEY
 TO CONTINUE"
440 IF INKEY$="" THEN GOTO 440
450 LET FA=(FA*M/100)+FA
460 LET C=0
470 LET Y(1)=NE*15/12:REM SALES
BASED ON NUMBER OF EMPLOYEES
480 LET Y(2)=(100-AS/FA)*MI/100:
REM SALES BASED ON MONTHLY INDUS
TRY SALES
490 REM NEXT LINES SET LOWEST FI
GURE FROM Y(1),Y(2),M(5) EQUAL T
O Y(3)
500 IF Y(1)<Y(2) AND Y(1)<M(5) T
```

```
HEN LET Y[3]=Y[1]:GOTO 540
510 IF Y[2]<Y[1] AND Y[2]<M[5] T
HEN LET Y[3]=Y[2]:GOTO 540
520 LET Y[3]=M[5]
530 REM NEXT LINES DETERMINE MON
THLY SALES
540 IF ABS[P1-AS]<501 THEN LET Y
[3]=3.6*Y[3]/3
550 IF Y[3]>M[5] THEN LET Y[3]=Y
[3]-1975:GOTO 550
560 MC=[MC*M/100]+MC
570 LET EF=Y[3]/M[5]*100:REM EFF
ICIENCY % AS SALES DIVIDED BY TO
TAL OUTPUT
580 LET AC=[MC*[ABS[85-EF]/3]/10
0]+MC:REM AVERAGE COST PER VEHIC
LE
590 LET MP=[[Y[3]*[AS-AC]]-[NE*A
W/12]]:REM MONTHLY PROFIT
600 LET MP=INT[MP/[100*1000]]
610 LET TP=TP+MP/10:REM TOTAL PR
OFIT IN MILLIONS
620 LET M=0
630 GOTO 40
640 REM ************************
*
650 REM REPORT PRINTOUT
660 CLS
670 PRINT "INDUSTRY SALES ";MI;"
 IN MONTH ";MT
680 IF MT>0 THEN PRINT "YOUR SAL
ES: ";INT[Y[3]];" [";INT[Y[3]*10
00/MI]/10;"% OF TOTAL]"
690 PRINT "-------------------------
-------------"
700 PRINT "YOU HAVE ";NE;" EMPLO
YEES"
710 PRINT "AVERAGE WAGES ARE $";
AW
720 PRINT " OR $M";INT[AW*NE/[10
0*1000]/12]/10;" PER MONTH"
730 PRINT "-------------------------
-------------"
740 IF MT=0 THEN RETURN
750 PRINT "AVERAGE COST PER VEHI
CLE  IS $";INT[AC]
760 PRINT "AND AVERAGE SELLING P
RICE IS $";INT[AS]
770 PRINT "SO THE AVERAGE PROFIT
 IS $";INT[AS-AC]
780 PRINT "OR $M";INT[[AS-AC]*Y[
3]/[100*1000]]/10;" PER MONTH"
790 PRINT "-------------------------
-------------"
800 PRINT "PROFIT FOR THE MONTH
IS $M";MP/10
810 PRINT "& TOTAL PROFIT TO DAT
E IS $M";INT[TP*10]/10
820 PRINT "-------------------------
-------------"
830 RETURN
840 REM ************************
*
850 REM MONTH REPORT
860 CLS
870 IF MT>0 THEN PRINT "YOUR MON
THLY SALES ARE ";INT[Y[3]]
880 PRINT "-------------------------
-------------"
890 PRINT "MAXIMUM MONTHLY OUTPU
T:"
900 PRINT TAB[3];"FACTORY 1: ";I
NT[M[1]]
910 PRINT TAB[3];"FACTORY 2: ";I
NT[M[2]]
920 PRINT TAB[3];"FACTORY 3: ";I
NT[M[3]]
930 IF SF=1 THEN GOTO 960
940 PRINT TAB[3];"FACTORY 4: ";I
NT[M[4]]
950 PRINT "-------------------------
-------------"
```

```
960 PRINT "TOTAL OUTPUT IS ";INT
(M(5))
970 PRINT "————————————————————
————————"
980 PRINT "EFFICIENCY LEVEL IS "
;INT(EF);"%"
990 RETURN
1000 REM ************************
**
1010 REM GET REPLIES
1020 LET A$=INKEY$
1030 IF A$<>"Y" AND A$<>"N" THEN
 GOTO 1020
1040 PRINT TAB(22);A$
1050 FOR J=1 TO 300:NEXT J
1060 RETURN
1070 REM ************************
**
1080 REM INCREASE OUTPUT?
1090 IF M(4)=0 THEN LET X=15:GOT
O 1110
1100 LET X=18
1110 PRINT "IT WILL COST $M";X;"
 TO EXPAND"
1120 PRINT TAB(8);"OUTPUT BY 1%"

1130 PRINT "————————————————————
————————"
1140 PRINT "HOW MANY % DO YOU WI
SH TO RAISE OUTPUT?"
1150 INPUT EP:IF EP<0 OR EP>100
THEN GOTO 1150
1160 LET M(5)=0
1170 FOR T=1 TO 4
1180 LET M(T)=M(T)+M(T)*EP/100
1190 LET M(5)=M(5)+M(T)
1200 NEXT T
1210 LET TP=TP-EP*X
1220 FOR T=1 TO 250:NEXT T
1230 GOTO 170
1240 REM ************************
**
1250 REM SALE OF FACTORY FOUR
1260 PRINT "FACTORY 4 IS VALUED
FOR SALE AT $M104"
1270 PRINT "YOU CAN'T REBUY IT L
ATER IF YOU SELL IT..."
1280 PRINT "DO YOU WANT TO SELL
(Y/N)?"
1290 GOSUB 1010
1300 IF A$="N" THEN GOTO 210
1310 LET TP=TP+104
1320 LET SF=1
1330 LET M(5)=M(1)+M(2)+M(3)
1340 LET M(4)=0
1350 GOTO 170
1360 REM ************************
**
1370 REM CHECK ON LOSSES
1380 IF MP>0 THEN LET SA=0:GOTO
1480
1390 LET SA=SA+1
1400 IF SA>11 THEN GOTO 1420
1410 GOTO 1480
1420 CLS:PRINT
1430 PRINT "YOU JUST MADE YOUR T
WELFTH MONTHLY"
1440 PRINT "LOSS IN A ROW.......
..........."
1450 PRINT TAB(6);"YOUR EMPLOYME
NT"
1460 PRINT TAB(6);"IS HEREBY TER
MINATED!!"
1470 STOP
1480 IF TP>=-250 THEN GOTO 1530
1490 CLS:PRINT
1500 PRINT "UNDER YOUR MANAGEMEN
T, THE COMPANY HAS"
1510 PRINT "LOST MORE THAN $M250
!"
1520 GOTO 1450
1530 IF TP>200 THEN GOTO 1570
1540 RETURN
```

```
1550 REM ************************** 1680 LET NE=12000:REM STARTING N
**                                  O EMPLOYEES
1560 REM SWEET SWEET SUCCESS!!!     1690 LET AW=22995:REM STARTING A
1570 CLS:PRINT                      VERAGE WAGE
1580 PRINT "WELL DONE! THE COMPA    1700 LET AC=11100:REM COST PRICE
NY HAS MADE MORE"                   /VEHICLE
1590 PRINT "   THAN $M200.    YO    1710 LET AS=12000:REM SELLING PR
U'VE BEEN MADE"                     ICE
1600 PRINT "       A MEMBER OF      1720 LET MI=50*1000:LET MC=10100
THE BOARD"
1610 FOR T=1 TO 600:NEXT T          1730 LET Y(3)=12500
1620 STOP                           1740 LET MS=25:LET EF=77:LET FA=
1630 REM ************************** 160:LET SF=0:LET MT=0
**                                  1750 FOR J=1 TO 5
1640 REM INITIALIZATION             1760 READ M(J)
1650 PAPER 1: INK 7: BORDER 7: C    1770 NEXT J
LS                                  1780 RETURN
1660 RANDOMIZE                      1790 DATA 8900,3250,2500,1625,16
1670 DIM M(5):DIM Y(5)              275
```

Gridiron

```
10 REM GRIDIRON
15 REM SPECTRUM VERSION
20 CLS
30 RANDOMIZE
40 GOSUB 70
50 GOTO 220
60 REM ***************************
70 BORDER INT(RND*8)
72 LET W=W+.5
73 FOR X=1 TO 700:NEXT X
80 RETURN
90 REM ***************************
100 PRINT A$;SA
110 PRINT B$;SB
120 RETURN
130 REM ***************************

140 IF Z$=A$ THEN LET Z$=B$:RETU
RN
150 LET Z$=A$:RETURN
160 REM ***************************

170 IF INKEY$<>"" THEN GOTO 170
180 PRINT "      > PRESS ANY KEY
<"
190 IF INKEY$="" THEN GOTO 190
200 PRINT TAB(20);"OK":RETURN
210 REM ***************************

220 REM INITIALIZE
222 PAPER 1: INK 7: CLS
225 LET W=1: LET SB=0: LET SA=0
230 DEF FNA(X)=INT(RND*X)+1
240 PRINT "ONE PLAYER OR TWO"
250 INPUT X
260 IF X<1 OR X>2 THEN GOTO 250
270 IF X=1 THEN LET VC=1:LET A$=
"SILICON COWBOYS":GOTO 300
280 PRINT "WHAT IS THE NAME OF T
HE HOME TEAM?"
290 INPUT A$:IF A$="" THEN GOTO
290
300 PRINT "AND THE NAME OF THE V
ISITING TEAM?"
310 INPUT B$:IF B$="" THEN GOTO
310
320 LET Z$=A$:LET NU=35
330 CLS
340 PRINT "THERE ARE ";INT(10*(6
0-(W/4)))/10;" MINUTES TO GO"
350 PRINT TAB(8);Z$" TO KICK OFF
"
360 PRINT "YOU ARE ON YOUR OWN "
;NU;" YARD LINE"
370 IF VC=1 AND Z$=A$ THEN GOSUB
 70:GOTO 400
380 PRINT "TO KICK OFF..."
390 GOSUB 170
400 LET A=FNA(20)+40
410 PRINT Z$;" HAVE..."
420 FOR X=1 TO A
430 PRINT TAB(X/3);"KICKED ";X;"
 YARDS": BEEP .04,X-20
440 NEXT X
450 LET NU=NU+X
460 GOSUB 70
470 PRINT "THE BALL IS CAUGHT!"
480 GOSUB 70
490 LET A=FNA(30)+10
500 FOR X=1 TO A
510 PRINT TAB(X/5);"AND RETURNED
 ";X;" YARDS"
520 NEXT X
530 LET NU=ABS(100-NU+X)
540 GOSUB 140
550 PRINT "-----------------------
---------------"
560 PRINT "THE BALL IS DOWN ON"
```

```
570 PRINT Z$;"'S ";NU;" YARD LIN
E"
580 IF Z$=A$ AND VC=1 THEN GOSUB
 70:GOTO 600
590 GOSUB 170
600 LET TG=10:LET D=0:LET SL=NU
610 IF W=60 OR W=180 THEN GOTO 2
010
620 IF W=120 THEN GOTO 2070
630 IF W=240 THEN GOTO 2140
640 CLS
645 IF LEN B$<6 THEN LET B$=B$+"
      "
650 PRINT "      "; PAPER 7; INK
 2; A$( TO 6);": "; BRIGHT 1;SA;
 BRIGHT 0;" —— ";B$( TO 6);": "
; BRIGHT 1;SB: PAPER 1: INK 7: P
RINT "————————————————————————
——————": BEEP .1,18
660 PRINT INT(10*(60-(W/4)))/10;
" MINUTES TO GO"
670 GOSUB 70
680 PRINT "————————————————————
————————"
690 PRINT Z$;" IN POSSESSION"
700 PRINT TAB(4);D;" DOWN"
710 PRINT TAB(4);TG;" YARDS TO G
O"
720 PRINT "————————————————————
————————"
730 PRINT "START AT ";SL;" YARD
LINE"
740 PRINT "NOW ON ";NU;" YARD LI
NE"
750 PRINT 100-NU;" YARDS TO TOUC
HDOWN"
760 PRINT "————————————————————
————————"
770 PRINT "ON THIS PLAY ";
780 IF Z$=A$ THEN PRINT A$;" CAN
":GOTO 800
790 PRINT B$;" CAN"
800 PRINT "EITHER 1 - THROW"
810 PRINT "       2 - CARRY"
820 PRINT "    OR 3 - PUNT"
830 LET P=0
840 IF Z$=A$ AND VC=1 AND D<3 TH
EN LET P=2:GOTO 900
850 IF Z$=A$ AND VC=1 AND TG<7 T
HEN LET P=2:GOTO 900
860 IF Z$=A$ AND VC=1 AND (100-N
U)<31 THEN LET P=3:GOTO 900
870 IF Z$=A$ AND VC=1 THEN LET P
=1:GOTO 900
880 LET K$=INKEY$:IF K$<"1" OR K
$>"3" THEN GOTO 880
890 LET P=VAL(K$):PRINT TAB(10);
"OK ";P
900 GOSUB 70
910 LET W=W+1
920 CLS
930 PRINT Z$;", YOUR QUARTERBACK
 HAS"
940 PRINT TAB(8);"GOT THE BALL"
950 PRINT "————————————————————
————————"
960 PRINT "WAIT FOR THE COUNT, "
;Z$;","
970 PRINT TAB(8);"THEN HIT ANY K
EY..."
980 IF INKEY$<>"" THEN GOTO 980
990 GOSUB 70
1000 GOSUB 2200
1010 IF E=11 THEN GOTO 2340
1020 PRINT "————————————————————
————————"
1030 IF P=1 THEN GOTO 1050
1032 IF P=2 THEN GOTO 1310
1036 IF P=3 THEN GOTO 1590
1040 REM **************************
*
1050 PRINT "YOU'VE THROWN ";E*5;
```

```
" YARDS"
1060 PRINT TAB(4);"AND THE PLAY
IS..."
1070 LET A=FNA(8)
1080 IF A=1 THEN GOTO 1520
1090 LET A=FNA(E+1)
1100 IF A=1 THEN PRINT TAB(20);"
...COMPLETE":GOTO 1220
1110 PRINT TAB(20);"...INCOMPLET
E":LET D=D+1
1120 GOSUB 170
1130 PRINT "----------------------
----------"
1140 IF D>3 THEN GOTO 1160
1150 GOTO 610
1160 PRINT "THAT WAS YOUR 4TH DO
WN"
1170 PRINT "AND YOU'VE LOST POSS
ESSION!!"
1180 LET D=0:LET TG=10:LET NU=AB
S(100-NU):LET SL=NU
1190 GOSUB 70
1200 GOSUB 140
1210 GOTO 610
1220 GOSUB 170
1230 LET NU=NU+(E*5):LET TG=TG-(
E*5)
1240 IF NU>100 THEN GOTO 1800
1250 IF TG<1 THEN GOTO 1280
1260 LET D=D+1:IF D>3 THEN GOTO
1160
1270 GOTO 610
1280 LET D=0:LET TG=10:LET SL=NU
1290 GOTO 610
1300 REM ************************
*
1310 LET A=FNA(15)
1320 IF A=1 THEN GOTO 1510
1330 LET E=A-5
1340 IF E<0 THEN GOTO 1440
1350 IF E=0 THEN LET E=1:GOTO 13
70
1360 PRINT "GOOD SNAP, PASS AND
RUN"
1370 PRINT "YOU'VE GAINED ";E;"
YARDS"
1380 GOSUB 170
1390 LET TG=TG-E:LET NU=ABS(NU+E
):LET D=D+1
1400 IF NU>100 THEN GOTO 1800
1410 IF TG<1 THEN GOTO 1280
1420 IF D>3 THEN GOTO 1160
1430 GOTO 610
1440 PRINT "GREAT RUNNING BY THE
 OPPOSITION HAS"
1450 PRINT "CAUSED YOU TO LOSE "
;ABS(E);" YARDS"
1460 LET TG=TG-E:LET NU=NU+E:LET
 D=D+1
1470 GOSUB 170
1480 IF D>3 THEN GOTO 1160
1490 GOTO 610
1500 REM ************************
*
1510 PRINT "BAD SNAP...YOU'VE"
1520 PRINT "FUMBLED...AND"
1530 PRINT "YOU'VE LOST POSSESSI
ON..."
1540 LET NU=100-NU:LET D=0:LET T
G=10:LET SL=NU
1550 REM ************************
*
1560 GOSUB 170
1570 GOTO 460
1580 REM ************************
*
1590 PRINT "NICE PUNT..."
1600 PRINT "YOU'VE KICKED"E*4"YA
RDS"
1610 LET NU=NU+E*4
1620 IF NU>100 THEN GOTO 1650
```

```
1630 PRINT "----------
----------"
1640 GOTO 460
1650 LET A=FNA(3)
1660 IF A>1 THEN GOTO 1740
1670 PRINT "BUT YOU'VE MISSED TH
E GOAL!!"
1680 IF NU-E*4<80 THEN LET NU=AB
S(100-(NU-E*4)):GOTO 1700
1690 LET NU=20
1700 LET D=0:LET TG=10:LET SL=NU

1710 GOSUB 140
1720 GOSUB 170
1730 GOTO 610
1740 PRINT ".....AND SCORED
!"
1750 IF Z$=B$ THEN LET SB=SB+3:G
OTO 1770
1760 LET SA=SA+3
1770 GOSUB 100
1780 GOSUB 170
1790 LET NU=35:GOTO 330
1800 CLS
1810 FOR X=1 TO 5
1820 PRINT TAB(X*2);"TOUCHDOWN!!
!"
1830 NEXT X
1840 IF Z$=A$ THEN LET SA=SA+6:G
OTO 1860
1850 LET SB=SB+6
1860 GOSUB 100
1870 PRINT "TO PLAY FOR EXTRA PO
INT"
1880 GOSUB 170
1890 PRINT "----------
----------"
1900 PRINT "THE BALL IS SNAPPED.
..PREPARE TOKICK!": BEEP .1,-12
1910 GOSUB 70
1920 GOSUB 2200
1930 IF E>9 THEN PRINT "YOU MISS
ED":LET NU=20:GOTO 1970
1940 PRINT "YOU SCORED...":LET N
U=35
1950 IF Z$=A$ THEN LET SA=SA+1:G
OTO 1980
1960 LET SB=SB+1:GOTO 1980
1970 GOSUB 140
1980 GOSUB 100
1990 GOSUB 170
2000 GOTO 330
2010 FOR X=1 TO 10
2020 PRINT TAB(2*X);"PERIOD OVER
"
2030 NEXT X
2040 GOSUB 100
2050 GOSUB 170
2060 GOTO 660
2070 FOR X=1 TO 10
2080 PRINT TAB(2*X);"HALF TIME"
2090 NEXT X
2100 GOSUB 100
2110 LET Z$=B$
2120 GOSUB 170
2130 LET NU=35:LET W=W+2:GOTO 33
0
2140 FOR X=1 TO 10
2150 PRINT TAB(2*X);"GAME OVER"
2160 NEXT X
2170 GOSUB 100
2180 STOP
2190 REM **********************
*
2200 LET E=0:LET X=10
2210 IF Z$=A$ AND VC=1 THEN PRIN
T "THIS ONE FOR ";A$:GOTO 2290
2220 LET E=E+1:LET X=X-1
2230 PRINT TAB(E);E
2240 FOR Y=1 TO X*1.5
2250 IF INKEY$<>"" THEN LET Y=X*
1.5+1:RETURN
```

```
2260 NEXT Y
2270 IF E=11 THEN RETURN
2280 GOTO 2220
2290 FOR E=1 TO FNA(7)+2
2300 FOR J=1 TO 30:NEXT J
2310 PRINT TAB(E);E
2320 NEXT E
2330 RETURN
2340 PRINT "TOO LATE!"
2350 PRINT "YOU'VE BEEN SACKED!"

2360 LET E=FNA(4)
2370 IF E=3 THEN GOTO 2430
2380 PRINT "AND LOST FIVE YARDS!
"
2390 LET TG=TG+5:LET D=D+1:LET N
U=NU-5
2400 GOSUB 170
2410 IF D>3 THEN GOTO 1160
2420 GOTO 610
2430 PRINT "AND LOST POSSESSION!
"
2440 LET D=0:LET NU=ABS(100-NU+5
):LET SL=NU:LET TG=10
2450 GOSUB 170
2460 GOSUB 140
2470 GOTO 610
```

Tennis

```
10 REM TENNIS
15 REM SPECTRUM VERSION
20 PAPER 4: INK 0: CLS
30 RANDOMIZE
40 LET AA=0:LET BB=0:LET T=0:LET
 KA=0
50 LET XA=0:LET YA=0:LET ZA=0
60 LET XB=0:LET YB=0:LET ZB=0
70 DEF FNA(X)=INT(RND*X)+1
80 INPUT "ONE HUMAN PLAYER OR TW
O";A
90 IF A<1 OR A>2 THEN GOTO 80
100 IF A=1 THEN LET A$="BJORNX":
LET VC=1
110 IF VC=1 THEN GOTO 160
120 PRINT "PLEASE ENTER A SIX-LE
TTER NAME"
130 INPUT "NAME OF FIRST PLAYER"
;A$
140 IF LEN(A$)<6 THEN LET A$=A$+
" ":GOTO 140
150 LET A$=A$( TO 6)
160 INPUT "NAME OF SECOND PLAYER
";B$
170 IF LEN(B$)<6 THEN LET B$=B$+
" ":GOTO 170
180 LET B$=B$( TO 6)
190 LET S=1:LET AA=1:LET BB=1
200 CLS
210 LET P$=A$:LET R$=B$
220 REM ************************
******
230 IF P$=A$ THEN LET R$=B$
240 IF P$=B$ THEN LET R$=A$
250 PRINT P$;" SERVING"
260 PRINT BRIGHT 1;"DO YOU WANT
TO SERVE 1 - FAST"
270 PRINT BRIGHT 1;"
      OR 2 - SLOW"
280 IF P$=A$ AND VC=1 AND SC=0 T
HEN LET KB=1:GOSUB 1720:GOTO 330
290 IF P$=A$ AND VC=1 AND SC=1 T
HEN LET KB=2:GOSUB 1720:GOTO 330
300 LET K$=INKEY$
310 IF K$<"1" OR K$>"2" THEN LET
 300
320 LET KB=VAL(K$)
330 PRINT:PRINT TAB(6);KB;TAB(10
);"> IT'S A ";
340 IF KB=1 THEN PRINT "FAST";
350 IF KB=2 THEN PRINT "SLOW";
360 PRINT " SERVE..."
370 GOSUB 1720
380 IF KB=1 THEN LET EB=FNA(3):G
OTO 400
390 LET EB=FNA(8)
400 IF EB=1 THEN GOTO 450
410 IF EB=3 AND SC=0 THEN GOTO 5
20
420 IF EB=3 AND SC=1 THEN GOTO 5
90
430 GOTO 670
440 REM ************************
******
450 CLS:PRINT
460 PRINT TAB(8);"....ACE...."
470 GOSUB 1720
480 LET SC=0
490 IF P$=A$ THEN GOTO 1140
500 GOTO 1150
510 REM ************************
******
520 CLS:PRINT
530 PRINT TAB(12);"....OUT...."
540 PRINT TAB(8);"...SECOND SERV
E..."
```

```
550 GOSUB 1720
560 LET SC=1
570 GOTO 230
580 REM ************************** ******
590 CLS:PRINT
600 PRINT TAB(12);"....OUT...."
610 PRINT TAB(9);"....DOUBLE FAU LT..."
620 GOSUB 1720
630 LET SC=0
640 IF P$=A$ THEN GOTO 1150
650 GOTO 1140
660 REM ************************** ******
670 LET SC=0
680 CLS:PRINT
690 IF INKEY$<>"" THEN GOTO 690
700 PRINT R$;", THE BALL IS":PRI NT "IN YOUR COURT"
710 PRINT "------------------------ -"
720 IF R$=A$ AND VC=1 THEN GOTO 750
730 PRINT "HIT ANY KEY, WHEN YOU  SEE THE     ZERO, TO RETURN THE  BALL..."
740 IF INKEY$<>"" THEN GOTO 740
750 LET X=4*FNA(3):LET Y=X
760 GOSUB 1720
770 LET E=5
780 PRINT TAB(2*(11-E));E
790 LET Y=Y-1
800 LET S$=INKEY$
810 IF S$<>"" AND E=0 THEN GOTO 890
820 IF S$<>"" THEN GOTO 990
830 IF Y>0 THEN GOTO 790
840 LET E=E-1:LET Y=X
850 IF E<-1 THEN GOTO 990
860 IF E=-1 AND R$=A$ AND VC=1 T HEN GOTO 890
870 GOTO 780
880 IF KB=1 THEN LET EA=FNA(2):G OTO 1000
890 LET EA=FNA(4)
900 IF E=0 AND R$=A$ AND VC=1 TH EN LET EA=FNA(8)
910 IF EA=1 THEN GOTO 940
920 IF R$=A$ THEN LET R$=B$:GOTO  670
930 LET R$=A$:GOTO 670
940 PRINT R$;", YOU'VE HIT THE B ALL"
950 PRINT TAB(8);"OUT OF PLAY... "
960 GOSUB 1720
970 IF R$=A$ THEN LET R$=B$:GOTO  1150
980 GOTO 1140
990 LET EA=FNA(4)
1000 IF EA=1 THEN GOTO 1070
1010 PRINT "YOU MISSED THE BALL,  AND..."
1020 GOSUB 1720
1030 PRINT "   IT WAS IN...BAD M ISTAKE"
1040 GOSUB 1720
1050 IF R$=A$ THEN LET R$=B$:GOT O 1150
1060 GOTO 1140
1070 PRINT "YOU MISSED THE BALL  AND..."
1080 GOSUB 1720
1090 PRINT "   IT WAS OUT..WELL  LEFT"
1100 GOSUB 1720
1110 IF R$=A$ THEN LET R$=B$:GOT O 1140
1120 GOTO 1150
1130 REM ************************** *******
```

```
1140 LET AA=AA+1:GOTO 1160
1150 LET BB=BB+1
1160 IF AA<5 AND BB<5 THEN GOTO
1230
1170 IF (BB>4 AND AA<4) OR (BB>4
 AND BB-AA>1) THEN LET AA=1:LET
BB=1:GOTO 1500
1180 IF (AA>4 AND BB<4) OR (AA>4
 AND AA-BB>1) THEN LET AA=1:LET
BB=1:GOTO 1440
1190 IF AA>4 AND AA>BB THEN LET
C$="ADV":LET D$="---":GOTO 1320
1200 IF BB>4 AND BB>AA THEN LET
D$="ADV":LET C$="---":GOTO 1320
1210 LET C$="{DEUCE":LET D$="{DE
UCE":GOTO 1320
1220 REM ************************
*******
1230 RESTORE
1240 FOR D=1 TO AA
1250 READ C$
1260 NEXT D
1270 RESTORE
1280 FOR D=1 TO BB
1290 READ D$
1300 NEXT D
1310 REM ************************
*******
1320 CLS
1330 PRINT "-----------------------
--"
1340 PRINT "          SET SET S
ET"
1350 PRINT "-----------------------
--"
1360 PRINT "            1   2
3  GAME"
1370 PRINT A$;"     ";XA;" ";YA;
" ";ZA;" ";C$
1380 PRINT B$;"     ";XB;" ";YB;
" ";ZB;" ";D$
1390 PRINT "-----------------------
--"
1400 GOSUB 1720
1410 IF T<>1 THEN GOTO 230
1420 FOR T=1 TO 25: BEEP .1,T: B
EEP .1,T+3: NEXT T:PRINT:PRINT "
    MATCH OVER    ":STOP
1430 REM ************************
*******
1440 CLS
1450 PRINT "GAME TO ";A$
1460 GOSUB 1720
1470 IF S=1 THEN LET XA=XA+1:LET
 C$="0":LET D$="0":GOTO 1560
1480 IF S=2 THEN LET YA=YA+1:LET
 C$="0":LET D$="0":GOTO 1580
1490 IF S=3 THEN LET ZA=ZA+1:LET
 C$="0":LET D$="0":GOTO 1600
1500 CLS
1510 PRINT "GAME TO ";B$
1520 GOSUB 1720
1530 IF S=1 THEN LET XB=XB+1:LET
 C$="0":LET D$="0":GOTO 1560
1540 IF S=2 THEN LET YB=YB+1:LET
 C$="0":LET D$="0":GOTO 1580
1550 IF S=3 THEN LET ZB=ZB+1:LET
 C$="0":LET D$="0":GOTO 1600
1560 IF (XA>5 AND XB<5) OR (XA<5
 AND XB>5) THEN GOTO 1630
1570 IF (XA>5 AND XA-XB>1) OR (X
B>5 AND XB-XA>1) THEN GOTO 1630
1580 IF (YA>5 AND YB<5) OR (YA<5
 AND YB>5) THEN GOTO 1630
1590 IF (YA>5 AND YA-YB<1) OR (Y
B>5 AND YB-YA>1) THEN GOTO 1630
1600 IF (ZA>5 AND ZB<5) OR (ZA<5
 AND ZB>5) THEN GOTO 1680
1610 IF (ZA>5 AND ZA-ZB>1) OR (Z
B>5 AND ZB-ZA>1) THEN GOTO 1680
1620 GOTO 1640
1630 LET S=S+1
```

```
1640 LET AA=1:LET BB=1
1650 IF P$=A$ THEN LET R$=A$:LET
 P$=B$:GOTO 1320
1660 LET P$=A$:LET R$=B$:GOTO 13
20
1670 REM ************************
*******
1680 LET T=1
1690 GOTO 1320
1700 REM ************************
*******
1710 REM DELAY
1720 BORDER INT (RND*8): PAUSE 2
00: BEEP .15,INT(RND*55)-20
1730 RETURN
1740 DATA "0","15","30","40"
```

Grand Prix

```
10 REM GRAND PRIX
20 GOSUB 2200:REM INITIALIZE
30 GOSUB 1190:REM CHOOSE TRACK
40 REM ****************************

50 REM MAJOR LOOP
60 GOSUB 120:REM PRINTOUT
70 GOSUB 280:REM ACCELERATION/CH
ECK
80 GOSUB 450:REM ENGINE/BRAKES
90 GOSUB 500:REM CORNER/POSITION

100 GOTO 60
110 REM ****************************
*
120 REM PRINTOUT
130 CLS
140 PRINT INK 0; BRIGHT 1;"  ENG
INE TEMP: ";INT(ENG*10)/10;"C.MA
X.200"
150 PRINT INK 0; BRIGHT 1;"BRAKE
 TEMPERATURE: ";INT(BRAK*10)/10;
"C.MAX.500"
160 PRINT INK 0; BRIGHT 1;" DIST
ANCE COVERED: ";INT(DIST*10)/10"
METRES"
170 PRINT INK 0; BRIGHT 1;"
             :" ;INT(DIST*100/RR)
/100;" LAPS"
180 PRINT INK 0; BRIGHT 1;"YOUR
POSITION IS ";INT(FP)
190 PRINT "-------------------
-----------"
200 PRINT INK 2;"    CURRENT SPE
ED: ";INT(SPEED*10)/10;"KPH"
210 PRINT INK 2;"
  : ";INT(SPEED*5.555)/10;" METR
ES PER MOVE"
220 PRINT "-------------------
-----------"
230 PRINT INK 3;"CORNER APPROACH
ING ";INT(APP);" METRES"
240 PRINT INK 3;"RECOMMENDED SPE
ED: ";C(C);" KPH"
250 PRINT "-------------------
-----------"
260 RETURN
270 REM ****************************
*
280 REM CHECK ACCELERATION AND F
ACTORS
290 X$=INKEY$:IF X$<>"Z" AND X$<
>"M" AND X$<>" " THEN GOTO 290
300 PRINT TAB(12);"OK"
310 LET X=0
320 IF X$="M" THEN LET X=SPEED/1
5
330 IF X$="Z" THEN LET X=-SPEED/
15
340 LET NUM=NUM+1:REM NUMBER OF
MOVES
350 LET SPEED=SPEED+X
360 IF SPEED<0 THEN LET SPEED=0
370 LET TRAV=SPEED*.5555:REM DIS
TANCE TRAVELLED
380 LET DIST=DIST+TRAV:REM TOTAL
 DISTANCE TRAVELLED
390 LET ENG=ENG+(X/2)+.07:IF ENG
<70 THEN LET ENG=70+RND*8:REM EN
GINE TEMP
400 IF X>0 THEN LET BRAK=BRAK*.9
:REM BRAKE TEMP FALLING; ACCELER
ATING
410 IF X<1 THEN LET BRAK=BRAK-(3
*X)-RND*3:REM BRAKE TEMP INCREAS
ING; BRAKING
420 IF BRAK<8 THEN LET BRAK=8+RN
D*8
```

```
430 RETURN
440 REM ************************
*
450 REM CHECK ENGINE/BRAKE TEMP
460 IF ENG>200 THEN PRINT "YOUR
ENGINE HAS OVERHEATED":GOTO 830
470 IF BRAK>500 THEN PRINT "YOUR
 BRAKES HAVE OVERHEATED":GOTO 83
0
480 RETURN
490 REM ************************
*
500 REM CHECK CORNERING SPEED AN
D FIELD POSITION
510 LET APP=APP-TRAV
520 IF APP>0 THEN RETURN
530 LET CRASH=0
540 IF SPEED>[C(C)*1.125] THEN L
ET CRASH=1:GOTO 690
550 IF SPEED>[C(C)*1.1] THEN GOT
O 690
560 LET PNT=PNT+100-[[C(C)*1.1]-
SPEED]:REM CORNERING POINTS
570 LET NC=NC+1:REM NUMBER OF CO
RNERS
580 LET CP=96-[PNT/NC]:REM CORNE
RING POSITION
590 LET AM=AM+A(C):REM AVERAGE N
UMBER OF MOVES ALLOWED
600 LET RP=NUM-AM:REM RACING POS
ITION: YOUR MOVES MINUS AVERAGE
MOVE
610 LET FP=[CP+RP]/2:REM FIELD P
OSITION IS AVERAGE OF CORNER & R
ACE POSITIONS
620 IF FP<1 THEN LET FP=1
630 LET C=C+1
640 IF C=WW THEN LET C=1:REM LAP
 OVER
650 LET APP=APP+D(C)
660 IF LAP*QQ=AM THEN GOTO 910:R
EM RACE OVER
670 RETURN
680 REM ************************
*
690 REM CRASHED
700 CLS
710 PRINT "YOU CORNERED AT ";INT
[10*SPEED]/10;" KPH"
720 PRINT "AND THE MAXIMUM SPEED
 WAS JUST ";C[C]
730 GOSUB 2330
740 PRINT "YOU SPIN OFF THE TRAC
K..."
750 GOSUB 2330
760 IF CRASH=1 THEN GOTO 830
770 PRINT "YOU'VE LOST 20 SECOND
S, BUT YOU  ARE ABLE TO REJOIN T
HE RACE"
780 LET NUM=NUM+10:LET SPEED=INT
[2*C[C]/3]
790 LET PNT=PNT+50
800 GOSUB 2330
810 GOTO 570
820 REM ************************
*
830 PRINT ".....AND CRASH!
!!!!"
840 PRINT "————————————————————
————————————"
850 PRINT "YOU ONLY COMPLETED ";
INT[10*DIST]/10;" METRES,"
860 PRINT "OR ";INT[DIST*100/RR]
/100;" LAPS AND AT THAT"
870 PRINT "STAGE WERE IN POSITIO
N ";INT[FP]
880 PRINT "————————————————————
————————————"
890 GOTO 1050
900 REM ************************
*
910 REM RACE OVER
```

```
920 CLS
930 LET EFLAG=1
940 FOR X=1 TO 20
950 PRINT TAB(X);"WELL DONE, ";A
$;"!!"
960 PRINT TAB(21-X);"WELL DONE,
";A$;"!!"
970 NEXT X
980 PRINT "-------------------
-------------------"
990 PRINT "YOU MANAGED TO LAST O
UT THE FULL ";LAP;" LAP RACE..."

1000 PRINT "-------------------
-------------------"
1010 GOSUB 2330
1020 PRINT "YOU FINISHED IN POSI
TION ";INT(FP)
1030 PRINT "AFTER STARTING IN 6T
H POSITION..."
1040 GOSUB 2330
1050 PRINT "YOUR AVERAGE SPEED W
AS ";INT(DIST*180/NUM)/100;" KPH
"
1060 GOSUB 2330
1070 IF RP<1 THEN LET RP=1
1080 IF CP<1 THEN LET CP=1
1090 PRINT "YOU WERE ";INT(ABS(R
P));"TH FASTEST ON STRAIGHTS,"
1100 PRINT "AND ";INT(ABS(CP))"T
H FASTEST ON CORNERS."
1110 PRINT:PRINT "PRESS 'S' FOR
SAME RACE, 'N'    FOR NEW RACE,
'E' TO END"
1120 I$=INKEY$:IF I$<>"S" AND I$
<>"N" AND I$<>"E" THEN GOTO 1120

1130 IF I$="E" THEN STOP
1140 GOSUB 2240
1150 RESTORE
1160 IF I$="S" THEN GOSUB 1490:L
ET LAP=L2AP:GOTO 60
1170 IF I$="N" THEN CLS:GOSUB 12
50:GOTO 60
1180 REM **********************
**
1190 REM NAME AND TRACK DATA
1200 INPUT "WHAT IS YOUR NAME, D
RIVER";A$
1210 PRINT
1220 FOR X=1 TO 3
1230 PRINT TAB(4*X);"OK, GOOD LU
CK, ";A$
1240 GOSUB 2330:NEXT X
1250 PRINT "**********************
******************"
1260 PRINT "WHICH RACE DO YOU WA
NT TO TAKE PART IN:"
1270 PRINT
1280 PRINT TAB(2);"BRITISH GRAND
 PRIX  2650MT  :1"
1290 PRINT TAB(2);"GERMAN  GRAND
 PRIX  1700MT  :2"
1300 PRINT TAB(2);"ITALIAN GRAND
 PRIX  2200MT  :3"
1310 PRINT TAB(2);"MONACO  GRAND
 PRIX  3100MT  :4"
1320 PRINT
1330 PRINT TAB(2);"ENTER A NUMBE
R (1 TO 4)"
1340 K$=INKEY$
1350 IF K$<"1" OR K$>"4" THEN GO
TO 1340
1360 LET GP=VAL(K$)
1370 PRINT "**********************
**********"
1380 PRINT TAB(5);"OK, THE ";
1390 IF GP=1 THEN PRINT "BRITISH
";
1400 IF GP=2 THEN PRINT "GERMAN"
;
1410 IF GP=3 THEN PRINT "ITALIAN
";
```

```
1420 IF GP=4 THEN PRINT "MONACO"
;
1430 PRINT " RACE"
1440 PRINT "**********************
*********"
1450 PRINT:PRINT "OVER HOW MANY
LAPS?"
1460 INPUT LAP:IF LAP<1 THEN GOT
O 1460
1470 LET LAP=INT(LAP+.5):LET L2A
P=LAP
1480 REM ************************
*
1490 REM BRITISH DATA
1500 LET SPEED=140
1510 FOR X=1 TO 9
1520 READ D(X):REM DISTANCE BETW
EEN CORNERS
1530 NEXT X
1540 DATA 800,400,250,200,250,30
0,100,100,250
1550 FOR X=1 TO 9
1560 READ C(X):REM RECOMMENDED M
AXIMUM CORNERING SPEED
1570 NEXT X
1580 DATA 150,90,175,200,200,90,
90,150,150
1590 FOR X=1 TO 9
1600 READ A(X):REM AVERAGE NUMBE
R OF MOVES ALLOWED BETWEEN CORNE
RS
1610 NEXT X
1620 DATA 8,4,2,2,2,2,1,1,2
1630 LET APP=800:LET WW=10:LET Q
Q=24:LET RR=2650
1640 IF GP=1 THEN RETURN
1650 REM ************************
*
1660 REM GERMAN DATA
1670 LET SPEED=85
1680 FOR X=1 TO 7
1690 READ D(X)
1700 NEXT X
1710 DATA 600,200,100,150,250,20
0,200
1720 FOR X=1 TO 7
1730 READ C(X)
1740 NEXT X
1750 DATA 90,175,120,90,200,200,
175
1760 FOR X=1 TO 7
1770 READ A(X)
1780 NEXT X
1790 DATA 6,2,1,2,2,2,2
1800 LET APP=600:LET WW=8:LET QQ
=17:LET RR=1700
1810 IF GP=2 THEN RETURN
1820 REM ************************
*
1830 REM ITALIAN DATA
1840 SPEED=108
1850 FOR X=1 TO 7
1860 READ D(X)
1870 NEXT X
1880 DATA 800,300,100,150,300,35
0,200
1890 FOR X=1 TO 7
1900 READ C(X)
1910 NEXT X
1920 DATA 120,90,90,150,200,120,
150
1930 FOR X=1 TO 7
1940 READ A(X)
1950 NEXT X
1960 DATA 8,3,2,1,3,3,2
1970 LET APP=800:LET WW=8:LET QQ
=22:LET RR=2200
1980 IF GP=3 THEN RETURN
1990 REM ************************
**
2000 REM MONACO DATA
2010 LET SPEED=162.5
```

```
2020 FOR X=1 TO 14
2030 READ D[X]
2040 NEXT X
2050 DATA 400,100,100,300,400,30
0,150,200,200,200
2060 DATA 150,150,200,250
2070 FOR X=1 TO 14
2080 READ C[X]
2090 NEXT X
2100 DATA 175,150,175,200,120,20
0,175,90,175,150
2110 DATA 150,175,120,150
2120 FOR X=1 TO 14
2130 READ A[X]
2140 NEXT X
2150 DATA 4,1,1,3,4,3,1,2,2,2
2160 DATA 1,2,2,2
2170 LET APP=400:LET WW=15:LET Q
Q=30:LET RR=3100
2180 RETURN
2190 REM ************************
**
2200 REM INITIALIZATION
2210 BORDER 6: PAPER 7: INK 1: C
LS
2220 RANDOMIZE
2230 DIM A[14]:DIM C[14]:DIM D[1
4]
2240 LET C=1:LET FP=6:LET PNT=0:
LET NC=0:LET CP=0
2250 LET AM=0:LET RP=0:LET APP=0
2260 LET NUM=0:REM NUMBER OF MOV
ES
2270 LET ENG=100:LET BRAK=10:LET
 TRAV=0:LET DIST=0
2280 LET EFLAG=0
2290 LET X=0
2300 RETURN
2310 REM ************************
**
2320 REM DELAY
2330 PAUSE 200:BORDER INT[RND*8]
: BEEP .1,INT[RND*50]-20
2340 RETURN
```

Flight Simulation

```
10 REM FLIGHT SIMULATION
15 REM SPECTRUM VERSION
20 LET RPT=0
30 LET LD=INT(RND*360)
40 DIM E$(1000):REM THIS HOLDS F
LIGHT RECORD
50 DIM A$(7,14):DIM C$(7,14):REM
 THESE ARRAYS HOLD HORIZON AND C
OMPASS OUTPUT
60 REM ******************
70 GOSUB 2320:REM INITIALIZE
80 IF CRASH=0 THEN GOSUB 820:REM
 HORIZON/COMPASS
90 GOSUB 500:REM PRINTOUT
100 IF CRASH=1 THEN GOTO END
110 IF LAND=1 AND UFLAG=1 THEN P
RINT "WELL DONE. A PERFECT LANDI
NG!!":GOTO END
120 IF LAND=1 AND UFLAG=0 THEN P
RINT "YOUR WHEELS ARE UP":GOSUB
1780:GOTO 90
130 LET T=AIRSPEED:LET STALL=0
140 LET X$=INKEY$
150 IF X$="R" THEN LET RPT=1:GOT
O 70
160 IF RPT=1 AND E$(CLOCK+1)=""
THEN LET RPT=0:GOTO 140
170 IF RPT=1 THEN LET X$=E$(CLOC
K+1)
180 IF X$="" THEN GOTO 140
190 IF CLOCK<999 THEN LET E$(CLO
CK+1)=X$
200 IF TAKEOV=1 THEN LET ELEVATE
=INT(ELEVATE+RND*2-RND*3)
210 IF AIRSPEED<3 THEN GOTO 290
220 IF X$="Q" THEN LET ELEVATE=E
LEVATE+5:LET EFLAG=5:IF ELEVATE>
60 THEN LET STALL=1
230 IF X$="A" THEN LET ELEVATE=E
LEVATE-5:LET EFLAG=-5:IF ELEVATE
<-70 THEN LET STALL=-1
240 IF STALL<>0 THEN GOSUB 1640
250 IF ALTIMETER<1 THEN GOTO 290
:REM PREVENTS DRAMATIC TURNS ON
THE GROUND
260 IF X$="Z" THEN LET WA=WA-.5:
LET ANG=ANG-6:IF WA<-3 THEN LET
WA=-3
270 IF X$="M" THEN LET WA=WA+.5:
LET ANG=ANG+6:IF WA>3 THEN LET W
A=3
280 LET ANG=INT(ANG+RND*2-RND*2)

290 IF X$=" " THEN LET AIRSPEED=
AIRSPEED+8.5
300 IF X$="." THEN LET AIRSPEED=
AIRSPEED-7
310 LET AIRSPEED=AIRSPEED-ELEVAT
E/5
320 IF UFLAG=1 THEN LET AIRSPEED
=AIRSPEED-1.5:LET FUEL=FUEL-.5
330 IF AIRSPEED<0 THEN LET AIRSP
EED=0
340 IF AIRSPEED>400 THEN LET AIR
SPEED=400
350 IF X$="1" AND UFLAG=0 THEN L
ET UFLAG=1:GOTO 370
360 IF X$="1" AND UFLAG=1 THEN L
ET UFLAG=0
370 LET FUEL=FUEL-(ABS(T-AIRSPEE
D)/10)-3.75
380 IF FUEL<1 THEN GOSUB 1780
390 IF TAKEOV=1 THEN GOTO 420
400 IF ELEVATE>10 AND AIRSPEED>4
5 AND AIRSPEED<60 AND UFLAG=1 TH
EN LET TAKEOV=1
410 IF TAKEOV=0 THEN LET ALTIMET
ER=0:GOTO 450
```

```
420 IF LAND=0 AND AIRSPEED<30 TH
EN LET ELEVATE=ELEVATE-5:LET ALT
IMETER=9*ALTIMETER/10
430 LET ALTIMETER=ALTIMETER+INT(
([ELEVATE+.1]*AIRSPEED)+EFLAG*AI
RSPEED/1000)/80
440 IF ALTIMETER<300 AND TAKEOV=
1 THEN LET ALTIMETER=ALTIMETER+A
IRSPEED/30+ELEVATE
450 IF ALTIMETER<0 THEN GOSUB 17
80:REM CRASH
460 REM CHANGE NEXT TWO LINES TO
 MAKE IT       EASIER (OR EVEN H
ARDER) TO LAND
470 IF ALTIMETER>15 AND AIRSPEED
>20 OR TAKEOV=0 THEN GOTO 80
480 IF ABS(ANG-LD)<13 OR ABS(ANG
+360-LD)<13 THEN LET LAND=1:GOTO
 80
490 REM **********************
500 REM PRINTOUT
510 CLS
520 PRINT "   HORIZON";TAB(20);"
HEADING"
530 LET EV=INT(ELEVATE/10)
540 IF EV>2 THEN LET EV=2
550 IF EV<-2 THEN LET EV=-2
560 IF EV<>0 AND TAKEOV=1 AND CR
ASH=0 THEN GOSUB 1920
570 PRINT ":--------------------:--
-------:"
580 FOR J=1 TO 7
590 PRINT INK 7: PAPER 1;": ";A$
(J);" :";C$(J);" :"
610 NEXT J
620 PRINT ":--------------------
-------:"
630 LET DIST=DIST+ABS((COS(ELEVA
TE))*AIRSPEED)/360
640 LET CLOCK=CLOCK+1
650 PRINT ":RANGE ";INT(DIST*10)
/10;" : TIME ";INT(CLOCK)/10;" :
 ";LD
660 PRINT ":--------------------
-------:"
670 PRINT ":AIRSPEED : ";INT(AIR
SPEED)
680 PRINT ": ";Q$( TO INT(AIRSPE
ED/20));">"
690 PRINT ":ALTIMETER: ";INT(ALT
IMETER);
700 IF ANG<0 THEN PRINT TAB(19);
360+ANG;" DEG."
710 IF ANG>=0 THEN PRINT TAB(19)
;ANG;" DEG."
720 LET MR=INT(ALTIMETER/30):IF
MR>20 THEN LET MR=20
730 PRINT ": ";Q$( TO MR);">"
740 PRINT ":FUEL     : ";INT(FUE
L)
750 PRINT ": ";Q$( TO 20-INT(FUE
L/750));">"
760 PRINT ":--------------------
-------:"
770 PRINT ":ELEVATION: ";ELEVATE
 ": ";:GOSUB 2210:PRINT U$
780 IF UFLAG=1 THEN PRINT ":";TA
B(5);"> UNDERCARRIAGE DOWN < :"
790 IF UFLAG=0 THEN PRINT ":";TA
B(6);"> UNDERCARRIAGE UP <   :"
794 BEEP .03,12:BEEP .03,18
800 RETURN
810 REM **************************
**
820 REM ASSIGN HORIZON/COMPASS
830 IF ABS(INT(WA+.5))=3 THEN GO
SUB 980
840 IF ABS(INT(WA+.5))=2 THEN GO
SUB 1070
850 IF ABS(INT(WA+.5))=1 THEN GO
SUB 1160
860 IF INT(WA+.5)=0 THEN GOSUB 1
250
```

```
870 REM NEXT TWO LINES USED TO G
RADUALLY STRAIGHTEN UP WINGS
880 IF WA>0 THEN LET WA=WA-.2
890 IF WA<0 THEN LET WA=WA+.2
900 IF WA>.2 THEN GOTO 1350
910 FOR Z=1 TO 7
920 LET M$(8-Z)=A$(Z)
930 NEXT Z
940 FOR Z=1 TC 7
950 LET A$(Z)=M$(Z)
960 NEXT Z
970 GOTO 1350
980 REM WA=3 OR -3
990 LET A$(1)="              * "
1000 LET A$(2)="            **  "
1010 LET A$(3)="          **    "
1020 LET A$(4)="        **      "
1030 LET A$(5)="      **        "
1040 LET A$(6)="    **          "
1050 LET A$(7)="**              "
1060 RETURN
1070 REM WA=2 OR -2
1080 LET A$(1)="                "
1090 LET A$(2)="              ** "
1100 LET A$(3)="            ***  "
1110 LET A$(4)="          ***    "
1120 LET A$(5)="        ***      "
1130 LET A$(6)="***             "
1140 LET A$(7)="                "
1150 RETURN
1160 REM WA=1 OR -1
1170 LET A$(1)="                "
1180 LET A$(2)="                "
1190 LET A$(3)="            *****"
1200 LET A$(4)="          ******"
1210 LET A$(5)="****            "
1220 LET A$(6)="                "
1230 LET A$(7)="                "
1240 RETURN
1250 REM WA=0
1260 LET A$(1)="                "
1270 LET A$(2)="                "
1280 LET A$(3)="                "
1290 LET A$(4)="****************"
1300 LET A$(5)="                "
1310 LET A$(6)="                "
1320 LET A$(7)="                "
1330 RETURN
1340 REM ************************
***
1350 REM ASSIGN COMPASS STRINGS
1360 LET F2=ANG-F1
1370 IF F2<0 THEN LET FA=INT((F2
+375)/30)
1380 IF F2>=0 THEN LET FA=INT((F
2+15)/30)
1390 IF FA=12 THEN LET FA=0
1400 LET C$(1)="   .N.  "
1410 IF FA=11 THEN LET C$(2)="
.@:.. ":GOTO 1450
1420 IF FA=0 THEN LET C$(2)="  .
.@.. ":GOTO 1450
1430 IF FA=1 THEN LET C$(2)="  .
.:@. ":GOTO 1450
1440 LET C$(2)="  ..:.. "
1450 IF FA=10 THEN LET C$(3)= "
.@ : ..":GOTO 1480
1460 IF FA=2 THEN LET C$(3)= " .
. : @.":GOTO 1480
1470 LET C$(3)=" .. : .."
1480 IF FA=9 THEN LET C$(4)=" W@
-X--E":GOTO 1510
1490 IF FA=3 THEN LET C$(4)=" W-
-X-@E":GOTO 1510
1500 LET C$(4)=" W--X--E"
1510 IF FA=8 THEN LET C$(5)=" .@
 : ..":GOTO 1540
1520 IF FA=4 THEN LET C$(5)=" ..
 : @.":GOTO 1540
1530 LET C$(5)=" .. : .."
1540 IF FA=7 THEN LET C$(6)="  .
@:.. ":GOTO 1580
```

```
1550 IF FA=6 THEN LET C$(6)="  .
.@.. ":GOTO 1580
1560 IF FA=5 THEN LET C$(6)="  .
.:@. ":GOTO 1580
1570 LET C$(6)="  ..:.. "
1580 LET C$(7)="   .S.  "
1590 IF ANG>360 THEN LET ANG=ANG
-360
1600 LET F2=ANG
1610 IF W>0 THEN LET W=W-.4
1620 IF W<0 THEN LET W=W+.4
1630 RETURN
1640 REM STALL/FALL
1650 IF STALL=-1 THEN GOTO 1710
1660 FOR J=1 TO 10
1670 PRINT TAB(J); FLASH 1; INK
2;"YOU HAVE STALLED!":INK 7
1680 NEXT J
1690 LET AIRSPEED=AIRSPEED/4
1700 RETURN
1710 FOR J=1 TO 10
1720 PRINT TAB(J); INK 2; FLASH
1; "UNCONTROLLED DIVE!!":INK 7
1730 PRINT TAB(21-J); FLASH 1; I
NK 2;"PULL UP!!":INK 7
1740 NEXT J
1750 LET ALTIMETER=4*ALTIMETER/5

1760 RETURN
1770 REM ************************
***
1780 REM CRASH
1790 LET CRASH=1
1800 LET ALTIMETER=0
1810 LET H$="** *C R**  A ** S*
H* !!*":REM 25 CHARACTERS LONG
1820 CLS:FOR J=1 TO 20
1830 PRINT TAB(J);"CRASH!"
1840 PRINT TAB(21-J); INK 2;"CRA
SH!"
1850 NEXT J
1852 FOR T=10 TO -20 STEP -1: BE
EP .05,T: NEXT T
1860 FOR J=1 TO 7
1870 LET G=INT(RND*11)+1
1880 LET A$(J)=H$(G TO G+13)
1890 NEXT J
1900 RETURN
1910 REM ************************
***
1920 REM ADJUST HORIZON
1930 LET G$="                ":REM
 14 SPACES
1950 RETURN
2200 REM ************************
2210 REM INPUT INTO COMMAND NAME

2220 LET U$="————————"
2230 IF X$=" " THEN LET U$="THRO
TTLE ON"
2240 IF X$="." THEN LET U$="THRO
TTLE OFF"
2250 IF X$="Q" AND ALTIMETER>0 T
HEN LET U$="CLIMB"
2260 IF X$="Q" AND ALTIMETER=0 T
HEN LET U$="NOSE UP"
2270 IF X$="A" THEN LET U$="NOSE
 DOWN"
2280 IF X$="Z" THEN LET U$="BANK
 LEFT"
2290 IF X$="M" THEN LET U$="BANK
 RIGHT"
2300 RETURN
2305 STOP
2310 REM ************************
2320 REM INITIALIZATION
2330 PAPER 0: INK 7: CLS
2335 DIM M$(14,14): LET W=0
2340 RANDOMIZE
2342 LET END=2305
2350 LET Q$="————————————————————
——":REM 21 CHARACTERS IN STRING
```

```
2360 LET UFLAG=1:REM UNDERCARRIA
GE: 1 - DOWN, 0 - UP
2370 LET EFLAG=0:REM CLIMB RATE
2380 LET ANG=0:LET TAKEOV=0:LET
LAND=0
2390 LET AIRSPEED=0
2400 LET DIST=0:REM DISTANCE COV
ERED 'RANGE'
2410 LET ALTIMETER=0
2420 LET ELEVATE=0:REM ANGLE OF
ELEVATION
2430 LET WA=0:REM 'WING ANGLE; U
SED IN HORIZON PRINTOUT
2440 LET FUEL=750:LET CRASH=0:LE
T F2=0:LET F1=0:REM FOR DIRECTIO
N CHANGE/COMPASS ROUTINE
2450 LET CLOCK=0:REM TIME
2460 LET X$=""
2470 RETURN
```

Further Reading

Ahl, D. H., *Computers in Mathematics: A Sourcebook of Ideas,* Creative Computing Press, Morristown, New Jersey, 1979

Ahl, D. H., *Computers in Science and Social Studies: A Sourcebook of Ideas,* Creative Computing Press, Morris Plains, New Jersey, 1983

Cross, M. & R. D. Gibson, M. J. O'Carroll, T. S. Wilkinson (eds.), *Modelling and Simulation in Practice,* Pentech Press, Plymouth, Devon, UK, 1979

Frazer, J. R., *Introduction to Business Simulation,* Reston Publishing Company, Reston, Virginia, 1977

Hartnell, T., *Exploring Artificial Intelligence on your Commodore 64,* Bantam Books, New York, 1985; Interface Publications Ltd., London, 1985

Packer, R. E., *The Investor's Computer Handbook,* Hayden Book Company, Inc., Rochelle Park, New Jersey, 1982

Rich, E., *Artificial Intelligence,* McGraw-Hill Book Company, New York, 1983

Roberts, N. & D. Anderson, R. Deal, M. Garet, W. Shaffer, *Introduction to Computer Simulation,* Addison-Wesley Publishing Company, Reading, Massachusetts, 1983

Simondi, T., *What If. . . ? Guide to Computer Modeling,* The Book Company, Los Angeles, California, 1983

BYTE magazine (a McGraw-Hill publication) devotes a major portion of its March 1984 issue to computer simulations.

Acknowledgements

The oxygen cycle (chapter two) is based on information from Hoyle, T. *The Last Gasp,* Sphere Books Ltd., London, UK, 1983

Details of the *Mind Monitor* and power station simulator (chapter one) are from *The Australian* newspaper, February 29, 1985

Information used to create the flight simulation program (chapter twenty-three) came from:

Birch, N. H. & A. E. Bramson, *Flight Briefing for Pilots, Volume 4,* Pitman Publishing, London, UK, 1970
Champion, P., *Glider Pilot,* Model and Allied Publications Ltd., Hemel Hempstead, UK, 1974
Flying Magazine (editors), *Flying Wisdom,* Van Nostrand Reinhold Company, New York, 1979

Material related to the robot simulation programs (chapters seven, eight and nine) can be found in:

Bonner, P., *Build Your Own Gladiator,* article in *Personal Software* magazine, December 1983, pp. 123-127 and 198.
Burnett, J. D., *Logo, an Introduction,* Creative Computing Press, Morris Plains, New Jersey, 1982
Peddicord, R. G., *Understanding Logo,* Alfred Publishing Company, Inc., Sherman Oaks, California, 1983
Robillard, M. J., *Advanced Robot Systems,* Howard W. Sams and Company, Inc., Indianapolis, Indiana, 1984
Siklossy, L., *Let's Talk Lisp,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976

Program authors:

SPACE LANDING - Tim Hartnell
MONTE CARLO - Tim Hartnell
SIMULTANEOUS EQUATIONS - Tim Hartnell
LIFE - Tim Hartnell

ROBOT LOGO and POINT DUTY ROBOT - Tim Hartnell
CONNECT FOUR - Anthony W. Pearson
QUEVEDO CHESS MACHINE - Tim Hartnell
WASHINGTON D.C. - Philip J. Coates
STOCK MARKET - Philip J. Coates
DETROIT CITY - Philip J. Coates
GRIDIRON - Philip J. Coates
TENNIS - Philip J. Coates
GRAND PRIX - Philip J. Coates
FLIGHT SIMULATION - Tim Hartnell

Apple program conversion by Robert Young; Commodore 64 conversions by Ross Symons, Spectrum+/Spectrum conversions by Clive Gifford.

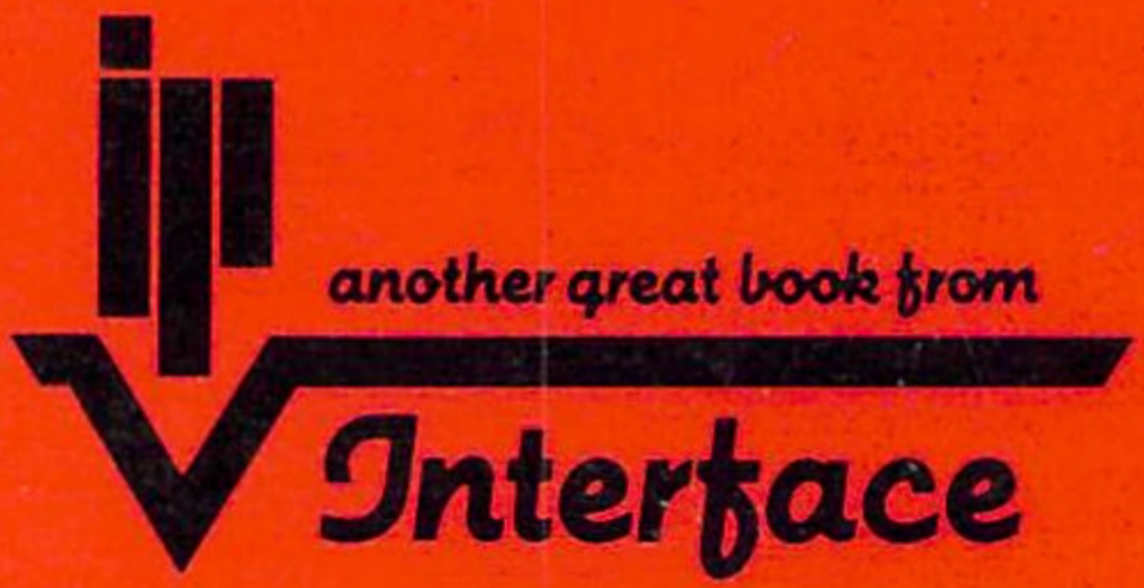
Imagine. Now you can use your computer as a window onto artificial reality. Fly a plane with FLIGHT SIMULATION. Control a ROBOT. Play the STOCK MARKET. Take part in a GRAND PRIX classic race, or participate in American GRIDIRON.

This book will give your computer power. Power to replicate reality in a way which can be quite uncanny.

From microbes which fight it out in a closed environment for the right to survive, to the program which puts you in the cockpit of a particularly temperamental plane, there is a wide range of programs to allow you to experience this power in action.

Of course, there is much more than program listings in this book. As well as the ready-to-run material, you'll discover how easy it is to analyse real-life situations so you can create your own computer simulations for entertainment and instruction.

Machine-specific listings included for SPECTRUM +, BBC MICRO, COMMODORE 64, AMSTRAD, all MSX machines, plus any machine which uses Microsoft BASIC.



ISBN 0-907563-80-5

