PRINT MAX 6 2 will give 6 as a result. Try writing a procedure to give the absolute value of a number, so that PRINT ABS 4 and PRINT ABS (-4) will both return the value 4.

Our game will ask you to type in your name and press Return. Here is a procedure to do this:

```
TO GET.NAME
   SPLITSCREEN
   PRINT1 [WHAT IS YOUR NAME?]
   MAKE "NAME FIRST REQUEST
   (PRINT "HELLO :NAME )
END
```

REQUEST waits for a line to be typed in and terminated with a Return. It then outputs the line as a list. FIRST outputs the first element of a list. Try the GET.NAME procedure and type in 'Holly' as the name. Now see what happens if 'Holly Johnson' is used as an input.

The game will control the turtle's onscreen movement by using the keys R, L and K. R will turn the turtle clockwise (right) through 30 degrees; L will turn it anticlockwise (left) by the same amount; while K is used to 'kick' the turtle — increasing its speed in whatever direction it is currently facing. The turtle will be moving around the screen, and we will require it to respond immediately to these keys. It would be a help if there was a LOGO primitive — READKEY, perhaps — that would output the last key that was pressed. If this was the case, we could write:

```
TO COMMAND
   MAKE "COM READKEY
   IF :COM = "R THEN RIGHT 30
   IF :COM = "L THEN LEFT 30
   IF :COM = "K THEN KICK
END
```

Unfortunately, this primitive does not exist! However, we can write it as a procedure, thus:

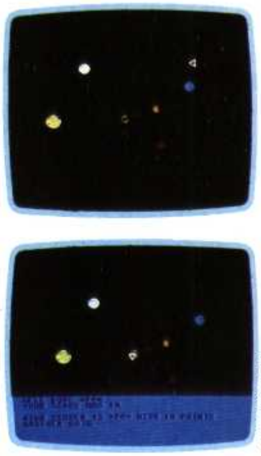
```
TO READKEY
   IF RC? THEN OUTPUT READCHARACTER
   OUTPUT "
END
```

When a key is pressed it is stored in the keyboard buffer. READCHARACTER simply outputs the last character from the buffer — if the buffer is empty READCHARACTER will wait for a key to be pressed and then output the relevant character. RC? is true if the buffer contains any characters and is false if the buffer is empty. So READKEY will now output the last character in the buffer, or will output a null word if the buffer is empty.

THE DYNATURTLE

Our space-going turtle is in fact a *dynaturtle*. This is a turtle that has a velocity, as well as a position and a heading like any normal earthbound turtle. The dynaturtle is in space, so there is no friction and no gravity. The dynaturtle will obey Newton's laws of motion. Our illustration will make this clearer, but as an example, let us assume that the dynaturtle is moving left to right across the screen with a velocity of 1. If the L key is pressed, the dynaturtle will turn to face the top of the screen, but the turtle's momentum will keep it moving on its horizontal course. If K is then pressed, the dynaturtle will get a 'kick' in the direction in which it is facing. This results in a push up the screen of velocity 1, and the dynaturtle will move diagonally across the screen with a velocity of 1.4. The dynaturtle will allow you to experiment with a body that obeys Newton's laws; it is designed to allow you to develop an intuitive understanding of these laws without you needing to understand all the relevant equations.

In the program, the dynaturtle's velocity is considered in terms of two components along the x and y axes. These components are found by using the SIN and COS functions. The only game controls are the three already mentioned. To begin the game, just type START. You have a fixed time in which to reach your goal, and the program keeps a record of the best score to date.



Extraterrestrial Turtle
The program as printed will generate the turtle and the target (its home base) only. The stars and planets shown here were added using some simple circle procedures

IAN McKINNELL

Logo Flavours

MIT LOGO	LCSI LOGO
DRAW	CS
PRINT1	TYPE
RC?	KEYP
READCHARACTER	RC
REQUEST	RL
SETHEADING	SETH
SETXY	SETPOS (followed by a list)
FULLSCREEN	FS (Not available on the Spectrum)
SPLITSCREEN	SS (Not available on the Spectrum)

IF has a slightly different syntax in LCSI LOGO, e.g:
IF DISTANCE < 5 [DONE STOP]

Program Project

Write a program to play Lunar Lander. For those who do not know this game, here is a brief rundown of the objectives:
You are piloting a rocket that is some distance above the Moon's surface, and which carries a limited amount of fuel. Gravity exerts a constant pull on your craft, and increases its downward velocity by a fixed amount each second. Pressing F causes your rocket engines to fire, slowing your craft's descent but burning up more fuel. The aim of the game is to use the F key so that your descent is slow enough for a safe landing.

Abbreviations

OUTPUT	OP
PRINT	PR
READCHARACTER	RC
REQUEST	RQ
SETHEADING	SETH