Teaching guide: input/output

This resource will help with understanding input/output. It supports Section 3.2.7 of our GCSE Computer Science specification (8525). The guide is designed to address the following learning aims:

- Review how the keyboard and the screen are the standard input and output devices.

Input/output (I/O)

Computer programs take inputs, process data and produce outputs. The inputs to a program can come from many locations: a microphone, a mouse, a motion sensor, a heat sensor, an input stream over a network, a digital camera, another file and so on. However, the 'standard' form of input is from a keyboard. Likewise, data can be output to many devices: speakers, printers, another file, actuators and so on but the 'standard' output is the monitor (or screen).

We have used two commands that enable data to come in or go out of a program:

USERINPUT

OUTPUT

Because the monitor and keyboard are considered standard input and output respectively, it is assumed that the data for **USERINPUT** comes from the keyboard and the data for **OUTPUT** is displayed on a screen.

Input

Every time **USERINPUT** is executed the program will wait for the user to type a sequence of characters on the keyboard and then, when the Enter or Return key is pressed, return that sequence of characters as a string to the program. Usually this is assigned to a variable, eg:

```
answer ← USERINPUT
```

which can then be used in the program, eg:

```
IF answer = 'YES' OR answer = 'yes' THEN
```

Note: you could not write this last line of code as:

```
IF USERINPUT = 'YES' OR USERINPUT = 'yes' THEN
```

Because this would make the user input two strings from the keyboard, not just the one.

Output

Every time the **OUTPUT** is executed the item or items following it are each evaluated, converted to a string representation, eg the expression `(10 * 2) / 8` would be evaluated to `2.5` (a real) and then the string `'2.5'` would be output to the screen. If there is more than one item they are separated by commas, and each one is output separately, eg

```
a ← 25
b ← 19
OUTPUT 'The result is ', (a * 4) / (b + 1)
```

would output

```
The result is 5
```