
LEVEL 3

I’m getting the hang of this

I now have choice in whether I do something or not ...

[again, the white rabbit hops its way towards a wall of green text ...]

New in Level 3:

Multiple functions, Nested Ifs, Variables

In this level variables are introduced. Variables can be created and updated with the following statements:

- **var** <Variable name> <**Boolean** | **Integer** | **Variable** | **String**>, this statement create a new variable with the name <Variable name> and assign the given value to it. Trying to create a variable that already created before will produce an error that will terminate the program
- **set** <Variable name> <**Boolean** | **Integer** | **Variable** | **String**>, this statement sets a value to an already created variable called <Variable name>. Trying to set a value to not created variable at the time of executing this statement will produce an error that will terminate the program

Where

- **Variable** refers to the current value an already created variable
- **String** is the same definition as last level, but now it cannot be a variable name

Variable will be used as values for the other possible statements defined in the previous level

- From now on, **if** / **else** statements can contain other **if** / **else** statements
- In this level there can be more than one function in the given code which all will be executed **independently**
- If a non boolean type was used in the condition of an if statement, it will produce an error and terminate the function. Non boolean in this context means something that can not be resolved to a boolean value e.g (integer, string that does not equal to true or false or a variable that does not contain a boolean value)
- If a function produced an error the string '**ERROR**' will be the resulting string
- Errors can only be produced when a statement is tried to be executed

Given a code as described above, run all the functions **independently** and output their resulting strings each in a new line.

The input format is identical to last level

	Input	Output
Format	N lineOfCode (repeated N times)	functionOutput (repeated N times)
Types	**N (int) number of code lines that follow** **lineOfCode (string) one or more space separated tokens**	**functionOutput (string) the output that print statements in the the given function produce**
Example	28 start print something var error initialised var error again end start print hello var hello world print hello var boolean false if boolean var hello old end else set hello boolean end print hello print world if true return true end else end print unreached end start print hello end	ERROR helloworldfalseworld hello


GOOD
LUCK!