Week 7: Python Programming - Consolidation

In this week's session we will focus on working with one larger program that can be extended in a number of ways. This enables lots of differentiation as students can design easier or hard amendments to the program.

```
import time

def maze():
    print("You are trying to find your way through a maze to the centre where ")
    print("there is a pot of gold!")
    print("What you don't know is that this is a dangerous maze with traps and hazards.")
    print()
    print("Starting maze game ...")
    print()
    print("You enter the maze...")
    time.sleep(2) # time.sleep is just used to build up the suspense!
    print("You reach a opening in the wall and go through it...")
    print()
    time.sleep(2)
    print("You can go left (L) or right (R)")
    answer = input("Make your choice ... ")
    print("You chose",answer,"... what will happen? ...")
    time.sleep(2)
    print("You turn the corner...")
    time.sleep(2)
    print("You walk forward a few steps...")
    time.sleep(2)
    if answer == "R":
        print("...and fall down a trap door and are never seen again....")
    else:
        print("...and see a beautiful grassy path lined with trees with a pot of gold at the end!")

    # end of program
```

Task 1: **Predict**

What will this program do? _____

_____

_____

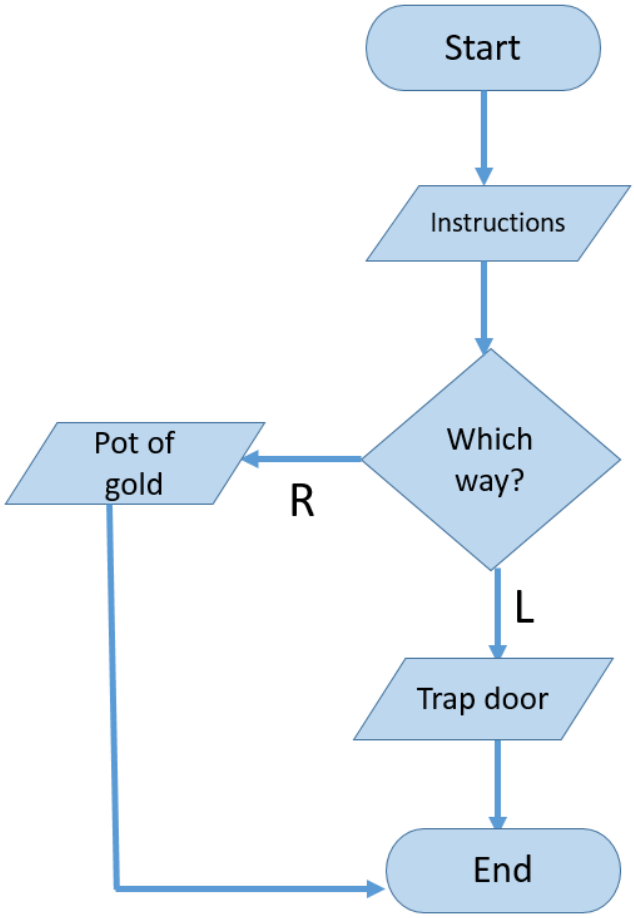
Task 2: Run

Download the starter program from http://teachinglondoncomputing.org/ks3-week-7

Run it and see if it does what you expected.

Task 3: In pairs, modify this program however you want. Design your changes in advance.

You may wish to draw a flowchart to demonstrate what your program will do. Here is an example.



Example of a flowchart for the maze starter program

Some suggested changes that you could build into the maze:

1. Ask the user if they want to go left, right or straight ahead.
2. Change the "pot of gold" option to another room and then have another option to exit by a blue door or a red door.
3. Use a loop to make sure that they can't continue until they put in one of the options.
4. Give the user a point every time they progress through one door and build in a few levels of doors and questions.
5. Change the scenario completely.

To help you with this task:

- Write down in advance, or draw in a flowchart, what you want your program to do.
- Identify if you will need any new variables to store data
- Keep earlier versions of your program in case you get in a muddle!