

Play Multiple Times

Problem ID: a13p02playmultipletimes

In the next version, the program should allow the user to play the game multiple times. Once the player obtains a victory, the user is asked if they want to play again. As long as the answer is affirmative, the game continues. Note that when the game is played again, the coin counter is reset, as well as resetting the position of the player to the original tile.

In this part, you need to make sure that your main program is short. It should only contain the main loop and a call to a `play()` function.

Input

This is similar to the previous problem of course, except for the added repetition. Updates to the input description as emphasized below.

The input consists of a sequence $l = (l_1, \dots, l_n)$ of n lines, where each line l_i should be interpreted as containing either an answer to the question where the player wants to move, an answer to the question whether the player wishes to pull a lever *or an answer to the question whether the player wants to play again*, depending on the state of the game at the time when that line is encountered.

In the tests, the last line of input will always correspond to the player *declining another round, after having reached the goal tile*. It is guaranteed that the player will reach the goal tile in at most 50 moves, *each time the game is played, and that the game will be played at most 10 times*.

Output

The quick way to see how the output should be, is to have a look at the given samples. But if you run into any problems, the following elaboration should help clarify.

More detailed output description

The output consists of multiple lines, which form a pattern consisting of a sequence of the following building blocks:

- A game block G , which can be further broken down into the basic building blocks as described in the previous problem.
- A repetition prompt P_R .
 - A repetition prompt just consists of the line “Play again (y/n): ”, without quotations. The space at the end is optional, as with the other prompts.

The sequence should alternate between those two, beginning with a game block, and ending with a repetition prompt (which will be declined).

After each repetition prompt, depending on the corresponding answer from the input, another game block should follow if the input is “y”, where the comparison should not be case-sensitive. Otherwise, the sequence ends, and there should be no further output.

Sample Input 1

```
n
y
e
y
w
n
n
e
y
e
s
n
s
y
n
y
n
e
y
e
s
n
s
n
```

Sample Output 1

```
You can travel: (N)orth.
Direction: Pull a lever (y/n): You received 1 coin, your total is now 1.
You can travel: (N)orth or (E)ast or (S)outh.
Direction: Pull a lever (y/n): You received 1 coin, your total is now 2.
You can travel: (S)outh or (W)est.
Direction: Pull a lever (y/n): You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (E)ast or (S)outh.
Direction: Pull a lever (y/n): You received 1 coin, your total is now 3.
You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: Pull a lever (y/n): You can travel: (N)orth or (S)outh.
Direction: Victory! Total coins 3.
Play again (y/n): You can travel: (N)orth.
Direction: Pull a lever (y/n): You received 1 coin, your total is now 1.
You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (E)ast or (S)outh.
Direction: Pull a lever (y/n): You received 1 coin, your total is now 2.
You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: Pull a lever (y/n): You can travel: (N)orth or (S)outh.
Direction: Victory! Total coins 2.
Play again (y/n):
```