10 The right change

4 points

Introduction

Until the time when everybody uses proximity cards and smartphones, we can still buy things in cash. Often times, we hand over a banknote whose value exceeds the total amount of the purchased items; the cashier will then hand us back the change in smaller notes or coins.

One traditional way of verifying that the change that we get back is correct is to start counting the change starting from the receipt's amount until we reach the note's value.

For instance, if we buy a book priced at 8.50€ with a 20€ note, the cashier may deliver the change as follows:

1. Here are 0.50€ that make up for 9€.
2. Here are 1.00€ that make up for 10€.
3. Here are 10.00€ that make up for 20€.

and we are good to go with our new book and the right change in our pocket.

This validation involves additions instead of subtractions, which are generally easier for humans.

You are asked to implement a computer program that, in this fashion, validates that the change returned back is correct.

Input

`Book's price` (an amount between 0.01 and 100.00€)
`Banknote` you are paying with (just one). Valid banknotes have the following value in Euros {5.00, 10.00, 20.00, 50.00, 100.00}.
`One line for every change iteration` (maximum 20), where each line includes the amount that the cashier is returning to us as a banknote or coin with any of the following values {0.01,0.02,0.05, 0.10, 0.20, 0.50, 1.00, 2.00, 5.00, 10.00, 20.00, 50.00}.
`0` ("0" in the input indicates that the casher is not giving us any more change)

Example 1

```
8.50
20.00
0.50
1.00
10.00
0
```

Example 2

```
7
10
1.00
0.5
1.00
0
```

Output

Output "`Right`" if the change is correct; output "`Wrong`" if the change is not correct. The order and the number of coins or banknotes returned by the cashier is not relevant as long as it matches the expected change.

Example 1
```
Right
```

Example 2
```
Wrong
```