Take-home Programming Exercises

NOTE: To be provided to the candidate.

Enclosed are *three* programming problems. We ask that you read all three descriptions Thoroughly, then create a program to solve only ***one*** of the problems. If you choose to do more than one, we will randomly choose and evaluate only one of your solutions, so it's better to work on only one of them.

While you have been given several days to complete the exercise, we realize you have a regular job and personal obligations. *Please don't spend more than four hours on the exercise.* We don't expect any more effort than that! Please create a project in your Github account and share the link with us beforehand. Please use the created project to complete the solution within three days or the alternative deadline offered by your interviewer. Please notify us when you complete. Request an extension, if you need it.

For the solution, we prefer that you use Python, but if you are more comfortable using another language, discuss this choice with your interviewer. You may not use any external libraries to solve this problem. However, any built-in "standard library" features, like parsing input arguments writing to the console, parsing strings, are fine.

However, you may use external libraries or tools for building or testing your code, as you see fit. For example, you might use `pytest` for Python code, `JUnit` for Java, etc. In fact, we encourage you to include tests! If you also execute your code using a command line invocation, for example by passing an argument list to control the behavior, please provide a "shell" script (or Makefile) that shows an example of this usage.

You may also include a brief explanation of your design and assumptions along with your code or use suitable code comments for this purpose.

Problem One: Mars Rover

A squad of robotic rovers are to be landed by NASA on a plateau on Mars. This plateau, which is flat and curiously rectangular, must be navigated by the rovers so that their on-board cameras can get a complete view of the surrounding terrain to send back to Earth.

A rover's position and location are represented by a combination of x and y co-ordinates and a letter representing one of the four cardinal compass directions. The plateau is divided up into a grid to simplify navigation. An example position might be 0, 0, N, which means the rover is in the bottom left corner and facing North.

In order to control a rover, NASA sends a simple string of letters. The possible letters are 'L', 'R' and 'M'. 'L' and 'R' makes the rover spin 90 degrees left or right respectively, without moving from its current spot. 'M' means move forward one grid point, and maintain the same heading.

Assume that the square directly North from (x, y) is (x, y+1).

1.1 Input

The first line of input is the upper-right coordinates of the plateau, the lower-left coordinates are assumed to be 0, 0.

The rest of the input is information pertaining to the rovers that have been deployed. Each rover has two lines of input. The first line gives the rover's position, and the second line is a series of instructions telling the rover how to explore the plateau.

The position is made up of two integers and a letter separated by spaces, corresponding to the x and y co-ordinates and the rover's orientation.

Each rover will be finished sequentially, which means that the second rover won't start to move until the first one has finished moving.

The output for each rover should be its final co-ordinates and heading.

Consider the following test input:

```
5 5
1 2 N
LMLMLMLMM
3 3 E
MMRMMRMRRM
```

Here is the expected output:

```
1 3 N
5 1 E
```

Problem Two: Sales Taxes

Basic sales tax is applied at a rate of 10% on all goods, except books, food, and medical products, which are exempt. An import duty is an additional sales tax applicable on all imported goods at a rate of 5%, with no exemptions.

When I purchase items, I receive a receipt that lists the name of all the items and their price (including tax), along with the total cost of the items, and the total amounts of the taxes paid. The rounding rules for sales tax are as follows; for a tax rate of n%, a shelf price of p contains (np/100 rounded up to the nearest 0.05) amount of sales tax.

Write an application that prints out the receipt details for these shopping Baskets:

Input Basket #1:

```
1 book at 12.49
1 music CD at 14.99
1 chocolate bar at 0.85
```

Input Basket #2:

```
1 imported box of chocolates at 10.00
1 imported bottle of perfume at 47.50
```

Input Backet #3:

```
1 imported bottle of perfume at 27.99
1 bottle of perfume at 18.99
1 packet of headache pills at 9.75
1 box of imported chocolates at 11.25
```

The calculator should produce the following output.

Output for Basket #1:

```
1 book: 12.49
1 music CD: 16.49
1 chocolate bar: 0.85
Sales Taxes: 1.50
Total: 29.83
```

Output for Basket #2:

```
1 imported box of chocolates: 10.50
1 imported bottle of perfume: 54.65
Sales Taxes: 7.65
Total: 65.15
```

Output for Basket #3:

```
1 imported bottle of perfume: 32.19
1 bottle of perfume: 20.89
```

```
1 packet of headache pills: 9.75
1 imported box of chocolates: 11.85
Sales Taxes: 6.70
Total: 74.68
```

Problem Three: Trains

The local commuter railroad serves several towns in Obscureland. Because of monetary concerns, all the tracks are 'one-way.' That is, a route from Hopelessfield to Tragicville does not imply the existence of a direct route from Tragicville to Hopelessfield and vise-versa. In fact, even if both of these routes exist, they are distinct and are not necessarily the same distance!

The purpose of this problem is to help the railroad provide its customers with information about the routes. In particular, you will compute the distance along a certain route, the number of different routes between two towns, and the shortest route between two towns.

The input will be a directed graph where a node represents a town and an edge represents a route between two towns. The weighting of the edge represents the distance between the two towns. A given route will never appear more than once, and for a given route, the starting and ending town will not be the same town.

For test input 1 through 5 shown below, if no such route exists, output 'NO SUCH ROUTE'. Otherwise, follow the route as given; do not make any extra stops! How long is the route in each case?

For example, the first problem means to start at city A, then travel directly to city B (a distance of 5), then directly to city C (a distance of 4).

Provide results for the following 10 cases:

1. The distance of the route A-B-C.
2. The distance of the route A-D.
3. The distance of the route A-D-C.
4. The distance of the route A-E-B-C-D.
5. The distance of the route A-E-D.
6. The number of trips starting at C and ending at C with a maximum of 3 stops. In the sample data below, there are two such trips: C-D-C (2 stops). and C-E-B-C (3 stops).
7. The number of trips starting at A and ending at C with exactly 4 stops. In the sample data below, there are three such trips: A to C (via B,C,D); A to C (via D,C,D); and A to C (via D,E,B).
8. The length of the shortest route (in terms of distance to travel) from A to C.
9. The length of the shortest route (in terms of distance to travel) from B to B.
10. The number of different routes from C to C with a distance less than

1. 30. In the sample data, the trips are: CDC, CEBC, CEBCDC, CDCEBC, CDEBC, CEBCEBC, and CEBCEBCEBC.

Test Input:

For the test input, the towns are named using the first few letters of the alphabet from A to D. A route between two towns (A to B) with a distance of 5 is represented as AB5.

```
Graph: AB5, BC4, CD8, DC8, DE6, AD5, CE2, EB3, AE7
```

Expected Output:

```
Output #1: 9
Output #2: 5
Output #3: 13
Output #4: 22
Output #5: NO SUCH ROUTE
Output #6: 2
Output #7: 3
Output #8: 9
Output #9: 9
Output #10: 7
```