Please read the instructions fully, then design and code a program that addresses the problem below. Please return your solution source files with instructions on how to run your example in a. zip file back to your recruiter.

We are interested in …

…production quality code.
…how good your Object Oriented Design skills are.
…how good your design pattern knowledge is.
…how good your unit tests are.
…how you solution will scale against new requirements in the future.

Venus Rover – AKQA – Developer Test

Introduction

Imagine we've been asked by NASA to develop the base for the control software for a team of robotic Venus Rovers. You need to write a program that allows NASA to send commands to the rovers. Remember to account for human errors. Also assume your code will be handed off to other developers for further development. You may write your code in any language you choose.

Requirements

The rover team must respond to a series of commands sent to them as consecutive lines.

Position
First, the rovers need to have an understanding of the flat plateau they are deployed to. The plateau center is at coordinate x=0, y=0. The position north of the origin is x=0, y=1.

Format:
```
Height North [space] Width East [space] Height South [space] Width West
Example: 5 5 3 3
```

Commands for each rover
Next, each rover is sequentially sent two separate lines of commands: Their starting position, and commands for movement.

Starting position format:
```
Starting X-coordinate [space] Starting Y-coordinate [space] Direction the rover is facing
Example: 1 2 N
```

Commands for movement format:
```
L = rotate the rover to the left (anti/counter clockwise) by 90 degrees
R = rotate the rover to the right (clockwise) by 90 degrees
M = move forward in the direction the rover is facing by 1 unit
Example: LMLMLMLMM
```

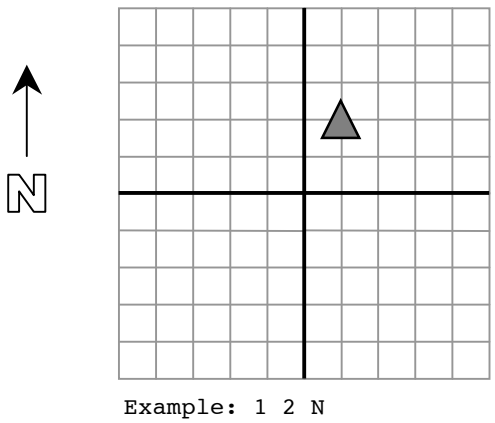
Other commands will be added in future so code accordingly. Bonus points for extra commands that yield interesting output data.

Output
The output from the rovers is expected to be their final position delivered in the same format as the starting position

Example input / output

Input
```
5 5 3 3
1 2 N
LMLMLMLMM
-3 -1 E
MMRMMLMMMR
```

Output
```
1 3 N
2 -3 S
```



Example: 1 2 N