SEAT:CODE

Technical Challenge

for SEAT:CODE Backend Developer

Welcome!

If you are reading this document, thanks for the effort invested so far. We feel you can be a great member of our family. We want to learn a bit from your technical background to carry on with the process.

Instructions

Problem

You will be facing a Kata alike problem. It's a code challenge small enough so it can be solved without investing too much time, but complex enough to give us some hints on your thinking process, problem-solving, testing, and software engineering skills.

Timing

The code challenge should take you around 4 hours. Feel free to invest more if you want to, but we understand that your time is a precious asset. We ask you to deliver your solution maximum 48 hours after you have received the challenge. So we can speed up the process and improve your candidate experience.

How to delivery the Code Challenge

Our preferred method of delivery is using GitHub. GitHub supports personal private projects, so it won't cost you any pence. We have created a Fake Kent Beck GitHub account (@kent-beck) who you will have to invite as a collaborator. If you don't know how to do it, take a look at "Inviting collaborators to a personal repository" Github Documentation Page. If for some reason you cannot use GitHub, you can zip your code challenge and send it to candidates@code.seat. Kent Beck won't be as happy, but it will do the trick. Send us also an email in case of any issue with the delivery.

Last words

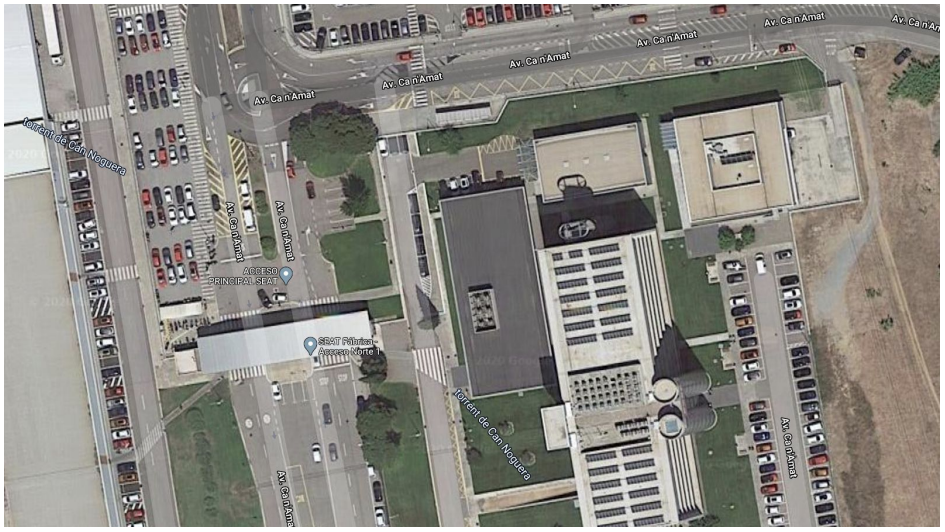
If you have any tech questions related to the challenge, take an assumption, and carry on. Please provide a small README file on how to compile and run your application. Best of luck! We hope you enjoy the code challenge!

Code Challenge

SEAT:CODE has been asked for a really important project. We need to develop an application that helps in controlling brand new mowers from the SEAT Martorell Factory. SEAT has rolled out brand new robotic mowers that are able to cut the grass and to inspect the terrain with their cameras to identify problems in the green areas.

Our robotic mowers are awesome!

SEAT Martorell factory has a lot of green spaces but for the MVP, we will consider only one single green grass plateau to simply the problem.



Picture from a small part of the SEAT Martorell Factory

A green grass plateau, which is curiously rectangular, must be navigated by the mowers so they can cut the grass and that their on-board cameras can get a complete view of the surrounding terrain to send to the SEAT Maintenance Office.

A mower's position and location are represented by a combination of X and Y coordinates and a letter representing one of the four cardinal compass points (N, E, S, W). The plateau is divided up into a grid to simplify navigation. An example position might be 0, 0, N, which means the mower is in the bottom left corner and facing North.

In order to control a mower, SEAT Maintenance Office sends a simple string of letters. The possible letters are "L", "R" and "M". "L" and "R" make the mower spin 90 degrees left or right respectively, without moving from its current spot. "M" means to move forward one grid point and maintain the same Heading.

Assume that the square directly North from (X, Y) is (X, Y + 1).

Input

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

The rest of the input is information pertaining to the mowers that have been deployed. Each mower has two lines of input. The first line gives the mower's position, and the second line is a series of instructions telling the mower 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 coordinates and the mower's orientation.

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

Output

The output for each mower should be its final coordinates and heading.

Input Test Case #1:
5 5
1 2 N
LMLMLMLMM
3 3 E
MMRMMRMRRM

Output Test Case #2:
1 3 N
5 1 E