

INSTRUCTIONS

A squad of robotic rovers are to be landed by NASA on a plateau on Mars.

This plateau, which is curiously rectangular, must be navigated by the rovers so that their onboard cameras can get a complete view of the surrounding terrain to send back to Earth.

Your task is to develop an API that moves the rovers around on the plateau.

In this API, the plateau is represented as a 10x10 grid, and a rover has state consisting of two parts:

- its position on the grid (represented by an X,Y coordinate pair)
- the compass direction it's facing (represented by a letter, one of N, S, E, W)

INPUT

The input to the program is a string of one-character move commands:

- L and R rotate the direction the rover is facing
- M moves the rover one grid square forward in the direction it is currently facing

If a rover reaches the end of the plateau, it wraps around the end of the grid.

OUTPUT

The program's output is the final position of the rover after all the move commands have been executed. The position is represented as a coordinate pair and a direction, joined by colons to form a string. For example: a rover whose position is **2 : 3 : W** is at square (2,3), facing west.

OBSTACLES

The grid may have obstacles. If a given sequence of commands encounters an obstacle, the rover moves up to the last possible point and reports the obstacle by prefixing **O :** to the position string that it returns. For instance, **O : 1 : 1 : N** would mean that the rover encountered an obstacle at position (1, 2).

EXAMPLES

- given a grid with no obstacles, input **MMRMMLM** gives output **2 : 3 : N**
- given a grid with no obstacles, input **MMMMMMMMMM** gives output **0 : 0 : N** (due to wrap-around)
- given a grid with an obstacle at (0, 3), input **MMMM** gives output **O : 0 : 2 : N**

RULES

- The rover receives a char array of commands e.g. **RMMLM** and returns the finishing point after the moves e.g. **2 : 1 : N**
- The rover wraps around if it reaches the end of the grid.