Goal

Build a program that implements the following mower specification.

Specification

The client, company X, wants to develop an algorithm to mow rectangular surfaces.

Mowers can be programmed to move throughout the entire surface. A mower's position is represented by Cartesian coordinates (X, Y), and an orientation (N, E, W, S). The lawn is divided into a grid to simplify navigation.

For example, the position 0,0,N indicates the mower is in the lower left corner of the lawn facing north.

To move the mower, we use a series of a combination of the following commands: L, R, and F. L and R turn the mower 90° left or right without moving the mower. F means move forward one space in the direction the mower is currently facing without changing its orientation.

If a forward movement would cause the mower to move outside of the lawn, the mower remains in the position and this command is discarded. The position directly to the north of (X, Y) is (X, Y + 1) and the position to the east of (X, Y) is (X + 1, Y).

Different mowers may not occupy the same space at the same time, and if a mower receives a move forward instruction that would cause it to run into another mower, the move forward instruction is silently discarded.

Your simulation will be run on a machine with multiple CPUs so multiple mowers should be processed simultaneously in order to speed up the overall execution time.

The mowers are programmed using an input file constructed in the following manner:

The first line corresponds to the upper right corner of the lawn. The bottom left corner is implicitly (0, 0).

The rest of the file describes the multiple mowers that are on the lawn. Each mower is described on two lines:

The first line contains the mower's starting position and orientation in the format "X Y O". X and Y are the coordinates and O is the orientation.

The second line contains the instructions for the mower to navigate the lawn. The instructions are not separated by spaces.

At the end of the simulation, the final position and orientation of each mower is output in the order that the mower appeared in the input.

When designing and implementing your solution ensure that you keep in mind separation of concerns, performance, and testing.

Example

Input file

```
5 5
1 2 N
LFLFLFLFF
3 3 E
FFRFFRFRRF
```

Result

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