

Auto Driving Car Simulation

You are tasked with developing a simulation program for an autonomous driving car, with the aim of competing with Tesla. Your team has already developed a prototype car, but it is still in its primitive stage.

The simulation program is designed to work with a rectangular field, specified by its width and height. The bottom left coordinate of the field is at position (0,0), and the top right position is denoted (width,height). For example, a field with dimensions 10 x 10 would have its upper right coordinate at position (9,9).

One or more cars can be added to the field, each with a unique name, starting position, and direction they are facing. For instance, a car named "A" may be placed at position (1,2) and facing North.

A list of commands can be issued to each car, which can be one of three commands:

- L: rotates the car by 90 degrees to the left
- R: rotates the car by 90 degrees to the right
- F: moves forward by 1 grid point

If a car tries to move beyond the boundary of the field, the command is ignored, and the car stays in its current position. For example, if a car at position (0,0) is facing South and receives an F command, the command will be ignored as it would take the car beyond the boundary of the field.

Users can interact with your simulation program through the command line interface. Upon launching the program, users are prompted with the following message:

```
Welcome to Auto Driving Car Simulation!
```

```
Please enter the width and height of the simulation field in x y format:
```

User is then able to enter:

```
10 10
```

The system responds with:

```
You have created a field of 10 x 10.
```

```
Please choose from the following options:
```

- ```
[1] Add a car to field
[2] Run simulation
```

User is then able to enter:

```
1
```

The system responds with:

```
Please enter the name of the car:
```

User is then able to enter:

```
A
```

The system responds with:

```
Please enter initial position of car A in x y Direction format:
```

User is then able to enter:

```
1 2 N
```

Please note that only N, S, W, E (representing North, South, West, East) are allowed for direction.

The system responds with:

```
Please enter the commands for car A:
```

User is then able to enter:

```
FFRFFFFRRL
```

This means car A will move forward twice, turn right, move forward four times, turn right twice, and turn left once.

The system responds with:

```
Your current list of cars are:
```

```
- A, (1,2) N, FFRFFFFRRL
```

```
Please choose from the following options:
```

```
[1] Add a car to field
[2] Run simulation
```

## Scenario 1 - Running simulation with a single car

At this point, there are a field, a car with initial position, facing, and commands available. If user attempts to run simulation, user can enter:

```
2
```

Then the system runs all the commands for car A, and responds with:

```
Your current list of cars are:
- A, (1,2) N, FFRFFFFRRL

After simulation, the result is:
- A, (5,4) S

Please choose from the following options:
[1] Start over
[2] Exit
```

If user chooses to start over, the system will show:

```
Welcome to Auto Driving Car Simulation!

Please enter the width and height of the simulation field in x y format:
```

If use choose to exit, the system will show:

```
Thank you for running the simulation. Goodbye!
```

## Scenario 2 - Running simulation with multiple cars

After user adds one car to the field, user can also choose to continue to add more cars. Hence, following example above, when system responds with:

```
Your current list of cars are:
- A, (1,2) N, FFRFFFFRRL

Please choose from the following options:
```

```
[1] Add a car to field
[2] Run simulation
```

User then can enter:

```
1
```

The system then responds with:

```
Please enter the name of the car:
```

User is then able to enter:

```
B
```

The system responds with:

```
Please enter initial position of car B in x y Direction format:
```

User is then able to enter:

```
7 8 W
```

The system responds with:

```
Please enter the commands for car B:
```

User is then able to enter:

```
FFLFFFFFFF
```

Please note that the length of commands do not have to be the same. If a car runs out of command, it will stay put.

The system responds with:

```
Your current list of cars are:
```

- A, (1,2) N, FFRFFFFRRL
- B, (7,8) W, FFLFFFFFFF

```
Please choose from the following options:
```

- [1] Add a car to field
- [2] Run simulation

At this point, user can continue to add more cars or run simulation. If user tries to add new car, the program follows the process above. If user tries to run simulation, then user will enter:

```
2
```

Then the system will run all car A's commands and all car B's commands, then respond with:

```
Your current list of cars are:
```

- A, (1,2) N, FFRFFFFRRL
- B, (7,8) W, FFLFFFFFFF

```
After simulation, the result is:
```

- A, collides with B at (5,4) at step 7
- B, collides with A at (5,4) at step 7

```
Please choose from the following options:
```

- [1] Start over
- [2] Exit

When processing commands for multiple cars, at every step, only one command can be processed for each car.

Using the example above:

- At step 1, car A moves forward, and car B moves forward.
- At step 2, car A moves forward, and car B moves forward.
- At step 3, car A turn right, and car B turns left.
- So on and so forth for the rest of the commands.

If some cars collide at certain step, then collided cars stop moving and no longer process further commands.

If cars do not have collision, then the system will print the final positions following example in Scenario 1.