

ActivePipe Coding Challenge

The Toy Robot challenge is well known coding challenge that can be solved in many different ways — we've given this one a twist. ActivePipe wants to use this challenge to assess your problem solving skills and gauge how you approach the problem.

A Github repository should be used to share your code. We want to see your focus on extensible and maintainable code. Implement this challenge as if you were working as a part of the ActivePipe team. Other team members may contribute to the codebase over time and the requirements may change.

The challenge is to simulate a toy robot moving on a square board of 6 x 6 units. The robot can roam around the surface of the board but shouldn't be able to fall off the edge. The twist is you are able to place any number of robots on the board and they move independently of each other. Any movement that would result in the robot falling from the board or colliding with another robot must be prevented, however further valid movement commands must still be allowed.

A robot, that we have named NAME in this example should read the following commands:

```
NAME: PLACE X,Y,F  
NAME: MOVE  
NAME: LEFT  
NAME: RIGHT  
NAME: REPORT
```

All robots should adhere to the following rules:

- Input can be from a file, or from standard input, as the developer chooses.
- PLACE will put a toy robot on the board in position X,Y and facing NORTH, SOUTH, EAST or WEST.
- The origin (0,0) can be considered to be the SOUTH WEST most corner.
- The first valid command for a robot is a PLACE command, after that, any sequence of commands may be issued, in any order, including another PLACE command. The application should discard all commands in the sequence for a given robot until a valid PLACE command has been executed
- MOVE will move a toy robot one unit forward in the direction it is currently facing.
- LEFT and RIGHT will rotate a robot 90 degrees in the specified direction without changing the position of the robot. REPORT will announce the X,Y and F of a robot. This can be in any form, but standard output is sufficient.
- A toy robot must not hit the edges of the board during movement, including when the robot is initially placed on the board.
- Any move that would cause the robot to fall off the edge of the board or collide with another robot must be ignored.
- Provide test data to exercise the application.

Examples

Example 1

```
ALICE: PLACE 0,0,NORTH  
ALICE: MOVE  
ALICE: REPORT
```

Expected output:

```
ALICE: 0,1,NORTH
```

Example 2

```
BOB: PLACE 0,0,NORTH  
BOB: LEFT  
BOB: REPORT
```

Expected output:

```
BOB: 0,0,WEST
```

Example 3

```
ALICE: PLACE 1,2,EAST  
ALICE: MOVE  
ALICE: MOVE  
ALICE: LEFT  
BOB: PLACE 3,3,EAST  
BOB: MOVE  
ALICE: MOVE  
ALICE: REPORT  
BOB: RIGHT  
BOB: MOVE  
BOB: REPORT
```

Expected output:

```
ALICE: 3,3,NORTH  
BOB: 4,2,SOUTH
```

Example 4

```
BOB: PLACE 1,3,SOUTH  
ALICE: PLACE 0,1,EAST  
ALICE: MOVE  
BOB: MOVE  
BOB: MOVE  
ALICE: MOVE  
BOB: MOVE  
BOB: LEFT  
ALICE: REPORT  
BOB: REPORT
```

Expected output:

```
ALICE: 2,1,EAST  
BOB: 1,1,EAST
```