Programming Exercise – Toy Robot

Description

The application is a simulation of a toy robot moving on a square tabletop, of dimensions 5 units x 5 units.

There are no other obstructions on the table surface.

The robot is free to roam around the surface of the table, but must be prevented from falling to destruction. Any movement that would result in the robot falling from the table must be prevented, however further valid movement commands must still be allowed.

Requirements

Create an application that can read in commands of the following form:

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

- Where PLACE will put the toy robot on the table 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.
- It is required that the first command to the 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 until a valid PLACE command has been executed.
- Where MOVE will move the toy robot one unit forward in the direction it is currently facing.
- Where LEFT and RIGHT will rotate the robot 90 degrees in the specified direction without changing the position of the robot.
- Where REPORT will announce the X,Y and F of the robot. This can be in any form, but standard output is sufficient.
- A robot that is not on the table can choose to ignore the MOVE, LEFT, RIGHT and REPORT commands.
- Input can be from a file, or from standard input, as the developer chooses.
- Provide test data to exercise the application.

Constraints

The toy robot must not fall off the table during movement. This also includes the initial placement of the toy robot. Any move that would cause the robot to fall must be ignored.

Deliverables

- The solution submitted should include structure, source code, configuration and any tests or test code you deem necessary - no need to package class files.
- Solve the problem in Java, C# or in a specific language that you may have been directed to use.
- Solve the problem as though it were "production level" code.
- It is not required to provide any graphical interface.

In order to get around firewall issues we recommend the solution be packaged as a password protected zip file.

Examples

Example 1

```
PLACE 0,0,NORTH
MOVE
REPORT

Output: 0,1,NORTH
```

Example 2

```
PLACE 0,0,NORTH
LEFT
REPORT

Output: 0,0,WEST
```

Example 3

```
PLACE 1,2,EAST
MOVE
MOVE
LEFT
MOVE
REPORT

Output: 3,3,NORTH
```