**Toy Robot**

**Scenario**

Create a library that can read in commands of the following form:

* PLACE X, Y, DIRECTION
* MOVE
* LEFT
* RIGHT
* REPORT

**Developing your Solution**

1. Use your preferred language, platform and IDE to implement this solution.
   * If C#: a console solution grounded in the Microsoft stack is ideal, but we're also happy to look at a solution with your preferred tech stack.
   * If iOS: please write an interactive UI for the application in Swift
   * If Android: please write an interactive UI for the application in Kotlin
2. Your solution should be clean and easy to read, maintain and execute.
3. You should provide build scripts or instructions to build and run the solution.
4. There should be a user interface to run the solution and assess that it works correctly. This could be a command prompt interface that takes one string command in at a time.
5. The code should be original and you may not use any external libraries or open source code to solve this problem, but you may use external libraries or tools for building or testing purposes.

**Solution Requirements**

* The library allows for a simulation of a toy robot moving on a 6 x 6 square tabletop.
* There are no 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 this must be prevented, however further valid movement commands must still be allowed.
* PLACE will put the toy robot on the table in position X,Y and facing NORTH, SOUTH, EAST or WEST.
* (0,0) can be considered as the SOUTH WEST corner and (5,5) as the NORTH EAST corner.
* The first valid 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 library should discard all commands in the sequence until a valid PLACE command has been executed.
* The PLACE command should be discarded if it places the robot outside of the table surface.
* Once the robot is on the table, subsequent PLACE commands could leave out the direction and only provide the coordinates. When this happens, the robot moves to the new coordinates without changing the direction.
* MOVE will move the toy robot one unit forward in the direction it is currently facing.
* LEFT and RIGHT will rotate the robot 90 degrees in the specified direction without changing the position of the robot.
* REPORT will announce the X,Y and orientation of the robot.
* A robot that is not on the table can choose to ignore the MOVE, LEFT, RIGHT and REPORT commands.

**Example Input and Output**

1. Example 1 Input

> PLACE 0,0,NORTH

> MOVE

> REPORT

Output: 0,1,NORTH

1. Example 2 Input

> PLACE 0,0,NORTH

> LEFT

> REPORT

Output: 0,0,WEST

1. Example 3 Input

> PLACE 1,2,EAST

> MOVE

> MOVE

> LEFT

> MOVE

> REPORT

Output: 3,3,NORTH

1. Example 4 Input

> PLACE 1,2,EAST

> MOVE

> LEFT

> MOVE

> PLACE 3,1

> MOVE

> REPORT

Output: 3,2,NORTH

**Code Complete**

When the solution is code complete, share your source code and supporting artefacts (e.g. GitHub/OneDrive/Google Drive) with the Consultant/ Talent Acquisition Specialist who contacted you from Telstra Purple.