



Toy Robot Code Challenge

The Toy Robot coding challenge is described in the following pages. Please review and complete an implementation that fulfills the requirements. Please take as much time as you need to complete the challenge. We would prefer you take the time to complete a solution you are happy with.

We use these submissions to help us understand your abilities and provide a discussion point should you be successful in progressing to the next stage. We are specifically interested in how you approach the problem. We like to see elegant solutions and clean code; write this as if someone else will be maintaining your code, with a preference on maintainability and correctness rather than performance.

Please complete the solution as a **C++** project of your choice.

What to include in your submission

- Link to hosted VCS repository (preferred), or zip of source code
- Appropriate documentation such as a readme (it should be clear how to setup and run your **C++** app)
- Appropriate unit and/or integration tests included with your source code (these should be runnable)

Description and requirements:

The application is a simulation of a toy robot moving on a square table top, 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.

Create a **console application** that can read in commands of the following form -

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

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. 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 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.

It is not required to provide any graphical output showing the movement of the toy robot.

The application should handle error states appropriately and be robust to user input.

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.

Example Input and Output:

a) -----

PLACE 0,0,NORTH

MOVE

REPORT

Output: 0,1,NORTH

b) -----

PLACE 0,0,NORTH

LEFT

REPORT

Output: 0,0,WEST

c) -----

PLACE 1,2,EAST

MOVE

MOVE

LEFT

MOVE

REPORT

Output: 3,3,NORTH