



After NASA's New Horizon successfully flew past Pluto, they now plan to land a Pluto Rover to further investigate the surface. You are responsible for developing the component that will allow the Rover to move around the planet. As you won't get a chance to fix your code once it is on-board, you are expected to use test driven development.

To simplify navigation, the planet has been divided up into a grid. The rover's position and location is represented by a combination of x and y coordinates and a letter representing one of the four cardinal compass points. An example position might be 0, 0, N, which means the rover is in the bottom left corner and facing North. Assume that the square directly North from (x, y) is (x, y+1).

In order to control a rover, NASA sends a simple string of letters. The only commands you can give the rover are 'F','B','L' and 'R'

- Implement commands that move the rover forward/backward ('F','B'). The rover may only move forward/backward by one grid point, and must maintain the same heading.
- Implement commands that turn the rover left/right ('L','R'). These commands make the rover spin 90 degrees left or right respectively, without moving from its current spot.
- Implement wrapping from one edge of the grid to another. (Pluto is a sphere after all)
- Implement obstacle detection before each move to a new square. If a given sequence of commands encounters an obstacle, the rover moves up to the last possible point and reports the obstacle.

Here is an example:

- Let's say that the rover is located at 0,0 facing North on a 100x100 grid.
- Given the command "FFRFF" would put the rover at 2,2 facing East.

Tips!

- Apply good software design principles.
- Keep your design simple, don't over-engineer.
- TDD all the way.
- Let the structure of your design evolve as you add more tests.
- Start simple. For instance, you might start with a test that if at 0,0,N with command F, the robot's position should now be 0,1,N.
- No user interface is required, the component should be implemented as a standalone library.
- Commit as you go, we like to see your progress and approach.
- Take no longer than 2 hours! Do what you can and give us a rough outline of what further changes you might consider making.