**Gaspedaal Assessment**

**Description of the problem**

Gaspedaal is moving into self-driving cars, and we started working on a software that takes care of the car navigation system in a 2D map.

Your task is to write a program that puts two cars in a rectangular 2D grid, receives movement instructions and returns the final location of these two cars.

The rectangular map is defined by two integer values [x y] defining its north-east corner. The south-west corner is defined by [0 0].

The starting position of each car is defined by its [x y] position and a letter indicating the direction it’s facing; N for north, W for west, S for south and E for east

The cars accept three kind of movement instructions: L, R and F:

L makes the car turn to the left.

R makes the car turn to the right.

F makes the car move one position forward.

Note that the turning commands do not move the car from its position, only turns it.

The movement instructions is list of any combination of these letters.

Inputs:

The program receives 5 inputs (arguments):

[x y] - Integer values. North-east corner position of the rectangular map

[x y z] - x and y are integers, z is a character. Starting position of the first car

[x y z] - x and y are integers, z is a character. Starting position of the second car

[string] - string composed by any combination of letters L, R and F defining the instructions to the first car.

[string] - string composed by any combination of letters L, R and F defining the instructions to the second car.

Assumptions:

Assume that the second car does not start moving until the first one finishes.

Assume that the second car starting position will never be the same as the first one.

**Steps to Run the application:**

1. To run the assessment :

**java -jar Gaspedaal\_run.jar 20 20 0 0 N 12 9 E FFFFFRFFRFLFF FFRFRFLFFLFFR**

1. To run the test cases :

**java -jar Gaspedaal\_unit\_test.jar**

**The application architecture:**

**The application is define 2 classes:**

1. Car class: that define the car with the positions X and Y, Direction (N, E, S, and W).
2. SelfDrivingCar: that has main function and all the important functions: movingCar and movingforward, changeDirectionLeft, changeDirectionRight.

**The Junit is defined with 5 Junit classes:**

1. carJunit : test the car class
2. SelfDrivingCarAccidentJunit : test if the car can make accident with another car or not
3. SelfDrivingCarDirectionJunit: test the directions that can be made from left and from right.
4. SelfDrivingCarForwardJunit: test the moving forward for the car.
5. selfDrivingCarmoveJunit: test the moving for the car according to the instructions.

**Assumptions that we made:**

1. Min x=0, min y=0, max x will be input and max y will be input.
2. The car will change the direction even if there is no forward after this direction.

For example: FFRFRFLFFLFFR, the car will turn to the right in the last command so the car direction after finished the instructions will be “S”.

1. If the car is moving forward and this movement will exceed the borders, the car will be stopped at the last forward and can’t proceed the remaining instructions.
2. If the car is moving forward and this movement will make an accident with another car, the car will be stopped at the last forward and can’t proceed the remaining instructions.