**Task**

**Objective**

The purpose of this test is to enable you to demonstrate your proficiency in solving problems using software engineering tools and processes.

Read the specification below and produce a solution. Your solution should be in the form of completed code.

The problem specified below requires a solution that receives input, does some processing and then returns some output. To capture the input we recommend a simple console application as developing a nice UI is not the objective of the test. You should provide sufficient evidence that your solution is complete by, as a minimum, indicating that it works correctly against the supplied test data. Using a unit testing framework would satisfy these requirements.

**Specification**

A large distribution company have just received a squad of prototype picking robots, which can be used to pick a given order from the stock within the warehouse by giving them a set of instructions. The company want to create a proof of concept that initially just shows that the robot can be navigated around an empty warehouse given a set of instructions. Navigating around objects, collisions, and actually picking the stock are out of scope for the initial development phase, you should just concentrate on moving the robots around the empty warehouse.

A robots position and direction can be represented by a combination of x and y co-ordinates and a letter representing one of the four cardinal compass points. The warehouse is divided up into a grid to simplify navigation. An example position might be 0, 0, N, which means the robot is in the bottom left corner of the warehouse and facing North.

In order to control a robot, the application needs to send a simple string of characters. The possible characters are '<', '>' and '^'. '<' and '>' make the robot spin 90 degrees left or right respectively, without moving from its current location.

Assume that the square directly North from (x, y) is (x, y+1).

'^' means move forward one grid point, while maintaining the same direction.

**Input**

The first line of input is the upper-right coordinates of the warehouse grid, the lower-left coordinates are assumed to be 0, 0.

The rest of the input is information pertaining to the robots that have been deployed. Each robot has two lines of input. The first line gives the robot’s position, and the second line is a series of instructions telling the robot how to navigate the warehouse.

The position is made up of two integers and a letter separated by spaces, corresponding to the x and y co-ordinates and the robot’s orientation.

Each robots movement will be finished sequentially, which means that the second robot won't start to move until the first one has finished moving. Collisions between robots are out of scope and do not need to be considered.

**Output**

The output for each robot should be its final co-ordinates and heading.

**Test Input:**

5 5

1 2 N

<^<^<^<^^

3 3 E

^^>^^>^>>^

**Expected Output:**

1 3 N

5 1 E