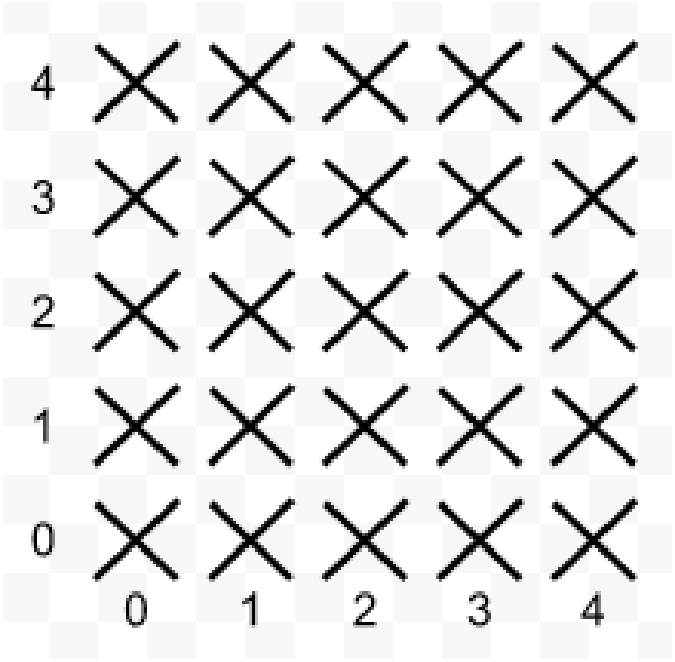
The interview problem

Problem description

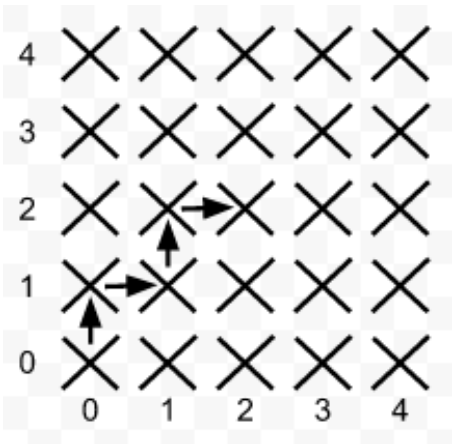
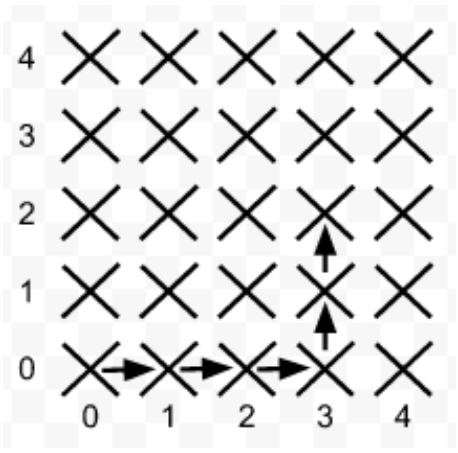
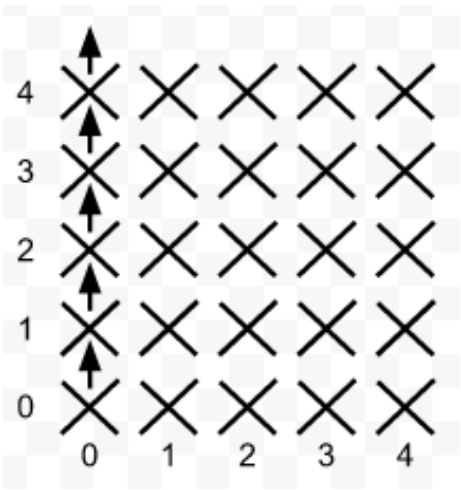
You have decided to implement a revolutionary new game! Realising that you only have a limited amount of time you have decided to implement it in separate stages. The first stage that you choose to create consists just of a playing board and a piece and the ability for a player to move the piece around the board. You decide to make the following initial assumptions and create a quick mock up of the board shown below.

- The board has a fixed size of 5 squares (assume it is always square).
- The piece can move around the board in one of four directions ((N)orth, (E)ast, (S)outh and (W)est).
- The piece will start in the bottom left corner of the board facing North.
- The bottom left board square is indicated by the position 0 0, top left corner of the board is 0 4, bottom right is 4 0 and top right is therefore 4 4.
- If you try to make a move that would result in the piece moving off the board the move will have no effect. For example moving North when you are already at the top of the board.
- You can assume that the input is always correct.



A player can issue a set of commands to the game which will result in an output of the location and direction of the piece after the moves. You choose to use the output format X Y Direction, so a piece that has moved two squares right and one north and is facing north would be in position 2 1 N. Commands are either a request to move forward (M) or to turn left (L) or right (R) by 90 degrees.

You come up with the following example inputs for a board of size 5.

Input	Output	Diagram	Notes
MRMLMRM	2 2 E		
RMMMLMM	3 2 N		
MMMMM	0 4 N		Moves to the top square and then attempts to make one further move north

We are looking for a solution that is **simple** but **well structured**. We only require a class library and some unit tests. No user interface (command-line or otherwise) is required. Use any object orientated language of your choice (preferably C#). Our preferred unit-test framework is NUnit, but feel free to use whichever you are most familiar with. The suggested duration of this exercise is 1-2 hours. When you're done, zip up the files that we'd need to build your code and run the tests, then email it back to us. We'll review it and if you come in for an interview, we'll discuss it with you and get you to extend/refactor it.