Coding Problem

We would like you to complete a small coding problem so we can gauge your development skills. You can assume that your code will need to be modified in the future by another developer, so structure it appropriately.

The assignment is to create a console application that reads a maze definition file, creates the appropriate data structures, and executes an algorithm for finding the exit point of a laser fired into the maze.

It should take the name of a definition file as input and display the results to the screen (details provided later).

We will test the executable with several different input files.

The Problem

You will be given a block of square rooms in an X by Y configuration, with a door in the center of every wall. Some rooms will have a mirror in them at a 45-degree angle. The mirrors may reflect off both sides (2-way mirrors) or reflect off one side and allow the beam to pass through from the other (1-way mirrors). When the laser hits the reflective side of one of the mirrors, the beam will reflect off at a 90-degree angle. Your challenge is to calculate the exit point of a laser shot into one of the maze. You need to provide the room it will be exiting through along with the orientation. The definition file will be provided through command line parameters.

The Mirrors

There are two types of mirrors that may appear in the definition file, 2-way and 1-way.

A 2-way mirror has a reflective surface on both sides. So no matter which side a beam strikes the mirror on, it will reflect off at a 90-degree angle away from the mirror.

A 1-way mirror has a reflective surface on one side. When a laser beam strikes the reflective side of the mirror, it will reflect off at a 90-degree angle away from the mirror. If the laser beam strikes the non-reflective side, it will pass through the room as if the mirror was not there.

The Definition File

The input file will be an ASCII text file with the following format:

```
The board size
-1
Mirror placements
-1
Laser entry room
-1
```

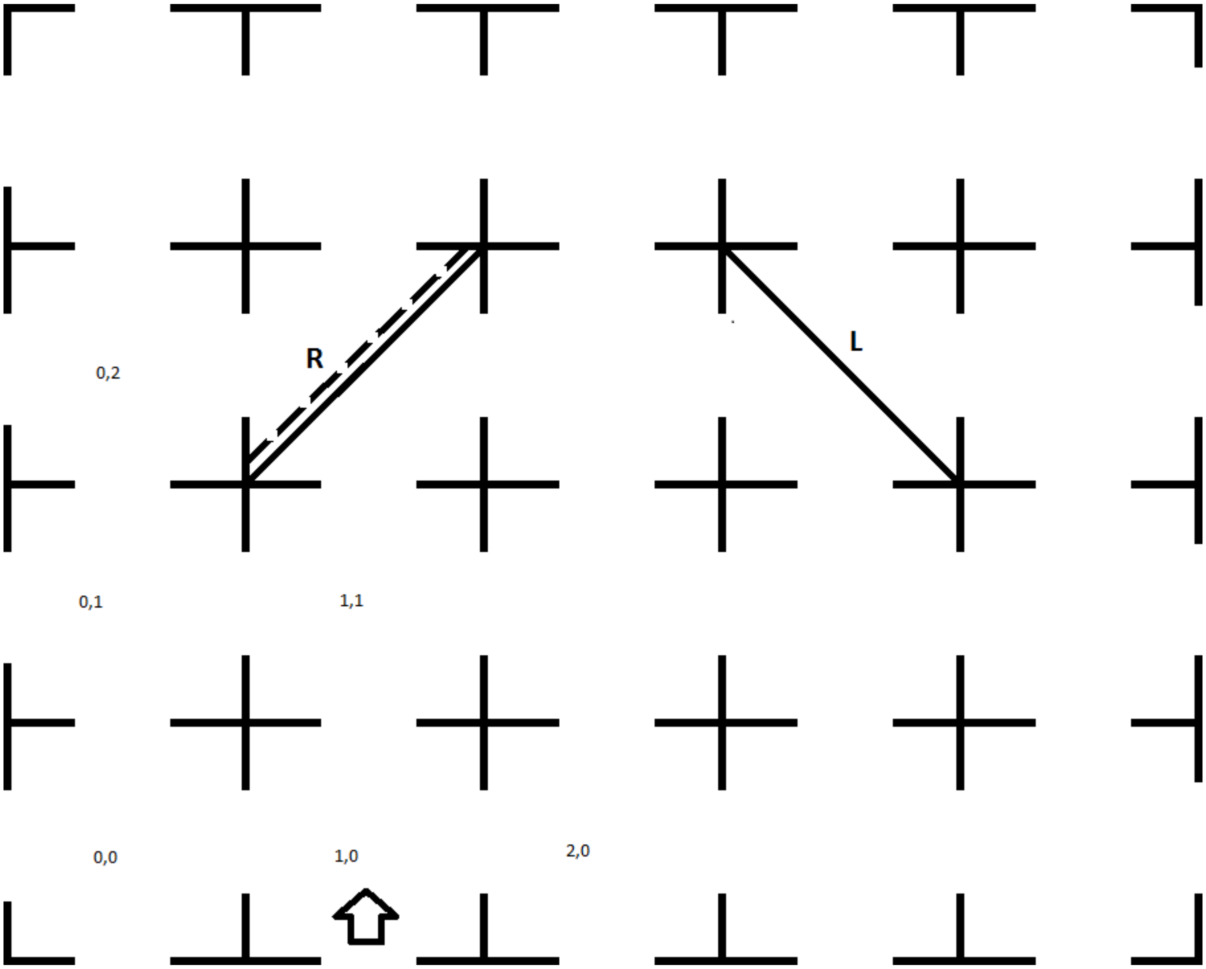
Description of each section of the definition file:

- The board size is provided in X,Y coordinates.
- The mirror placement will be in X,Y coordinates indicating which room the mirror is located. It will be followed by an R or L indicating the direction the mirror is leaning (R for Right and L for Left). That will be followed by an R or L indicating the side of the mirror that is reflective if it’s a 1-way mirror (R for Right Side or L for Left Side) or nothing if both sides are reflective and it’s a 2-way mirror.
- The laser entry room is provided in X,Y coordinates followed by an H or V (H for Horizontal or V for Vertical) to indicated the laser orientation.

A Sample Text File

```
5,4
-1
1,2RR
3,2L
-1
1,0V
-1
```

Visualization of the Sample Text File



Using the sample above, a laser starting at 1,0 would bounce off the mirror at 1,2 and 3,2 and exit the board at 3,0 vertically.

Another example: Let's use the same board size and mirrors but move the laser and start it at 0,2 and shooting horizontally. It would pass through the mirror at 1,2 and bounce off the mirror at 3,2 and exit the board at 3,0 vertically.

Output

At a minimum, your application should print the following to the screen:

1. The dimensions of the board
2. The start position of the laser in the format (X, Y) and the orientation (H or V)
3. The exit point of the laser in the format (X, Y) and the orientation (H or V)

Programming Languages

Write the code using an object oriented language such as C#, C++, Python, Java, etc. Choose one you are most comfortable with.

Submission

Submit your project using GitHub or Bitbucket. Your submission should include your code files, relevant scripts needed to compile your code (if there are any), and all the ancillary binary files required to run your application from a single directory on a Windows (7 or 8) machine. Frameworks, such as the .NET Framework or the Java Runtime Environment, are excluded from this.

You project needs to be completed, and a link to the repository should be emailed to us, at least 24 hours prior to your scheduled interview. Please ensure the repository is public, so we can access the code. If you can provide it earlier, please do so. If you have issues submitting through GitHub or Bitbucket, please contact us immediately.

Final Thoughts

We expect this exercise will probably take between 4 and 20 hours to complete. If you have any questions, please feel free to contact us. If you are unable to finish this prior to the 24-hour cutoff, please submit what you have finished and what you have working.