Coding Problem

We would like you to complete a small coding problem so we can gauge your coding and application structuring skills. You can assume that your code will need to be modified in the future by another developer.

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

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 application 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 open doors. 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 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

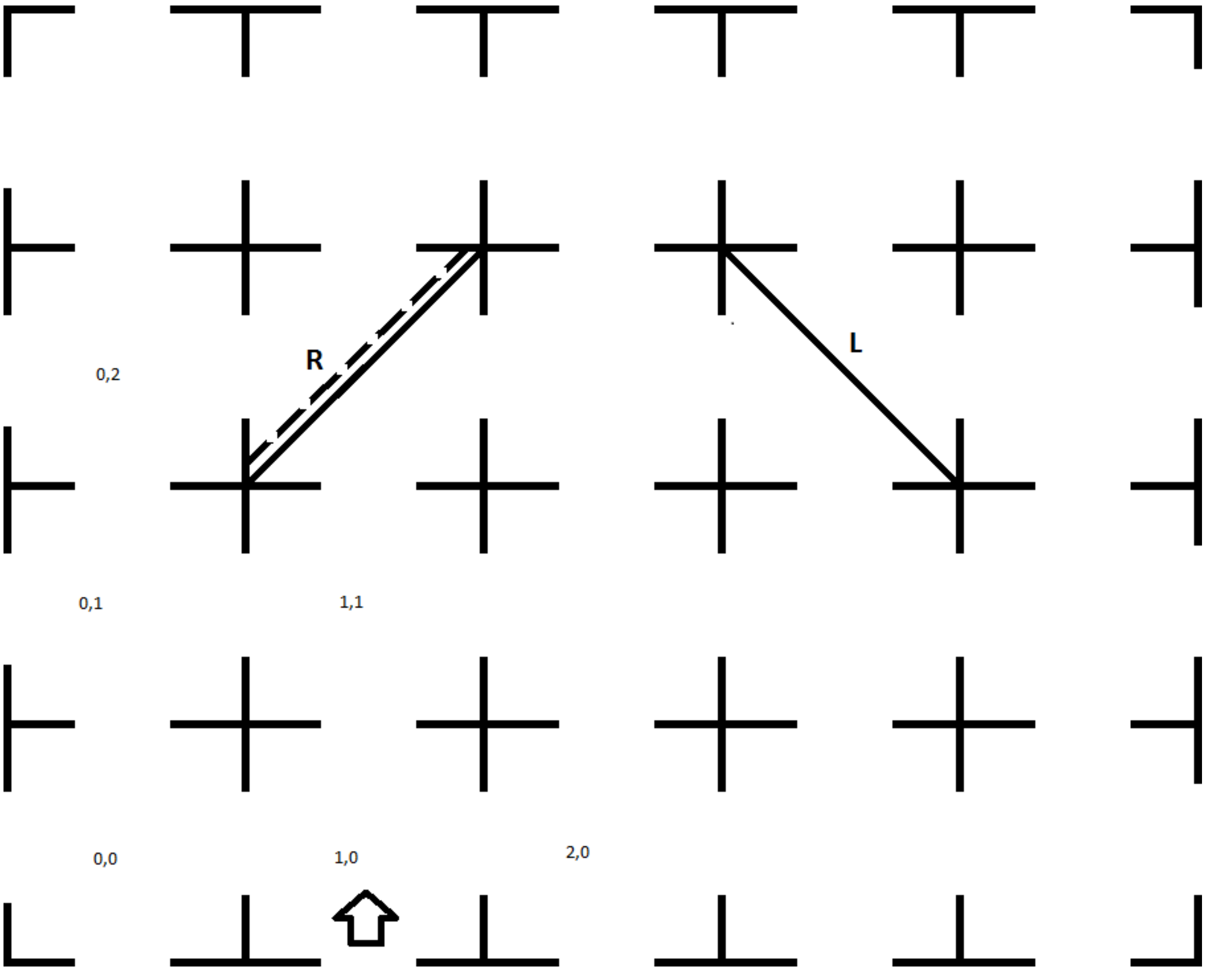

Figure 1 - Visual Example of Maze Structure and Mirror Placements

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)

Fell free to add additional functionality or testing you deem necessary.

Programming Languages

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

Submission

A link to a Git repository on a Git hosting service (Github, Gitlab, BitBucket) containing application code. The repository should contain all the necessary files to execute the application and a README containing instructions on how to do it.

The application must be provided at least 24 hours prior to your scheduled interview. If you can provide it earlier, then please do. The application should be executable in **UNIX based** environment.

If you are unable to host the application code on a Git hosting service, provide a ZIP file containing the executable, all code required to run the application and a README.

NOTE: When sending the email, please change the extension of the ZIP file from ZIP to TXT. If you do not make this change, our company message filters will not allow the message through. Contact us if you have issues.

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.