Goal

Create a command line application that builds and solves a simple “laser maze”. Please write original code, and do not reference existing solutions online. By accepting this coding challenge, you implicitly agree to not post or share solutions online, nor accept help from other engineers.

You will have 72 hours from time of receipt to complete and submit your solution. We expect this assignment to take between 1-3 hours, depending on the candidate’s skill level. If you anticipate this timeframe won’t be feasible, please reach out to OneLogin so we can do our best to accommodate special circumstances.

What is a Laser Maze?

A laser maze is a grid of squares, with a single solution. The player has a starting position from which he fires a laser in a specific direction (north, south, east, or west). The laser beam travels through the grid one square at a time. A square can be one of the following different “states”:

- Player starting position
  - this is the square where the maze begins
  - a beam will pass through a square in this state as if it were empty
- Empty
  - the laser beam passes through an empty square in the same direction as it entered
- Mirror
  - changes the direction of the laser beam. For instance, a mirror represented by “\” would change a beam traveling east to a beam traveling south. A mirror represented by “/” would change a beam traveling north to traveling east.

“Solving” the maze means letting the laser beam travel through the grid until it hits a wall, or gets stuck in a loop. The “solution” is two parts. The first part is the number of squares traversed.

Requirements

Command Line Application

The command line application can be run as follows:

```
maze ./path/to/input/file ./path/to/output/file
```

Input File

Note that the `# comments` in the examples below are for explanation only, and MUST NOT be present in your input and output files.

The input file must always be in the following format, single space as a delimiter and `\n` as a newline:

```
5 6        # represents grid size in X width and Y height
1 4 S      # represents player coordinate as X Y and laser direction
3 4 /      # represents coordinate for a mirror
3 0 /      # represents coordinate for a mirror (up to N mirrors)
1 2 \
3 2 \
4 3 \
```

Other useful information about the input file:

- It will always be of a correct format
- The grid size will be at least 1x1 and will be at most 1,000x1,000.
- There will be between 0 to 1,000 mirrors.
- There will never be two mirrors or a player that occupy the same space
- Player and Mirrors will never be outside of the grid
- Laser starting direction will be one of `S N E W` for South, North, East and West.
- There will be at minimum two lines in the input file
- Though the sample input file shows integers between 0-9, they may be as large as 1000.
- Coordinates use the 2D cartesian coordinate system from the 1st quadrant.

Output File

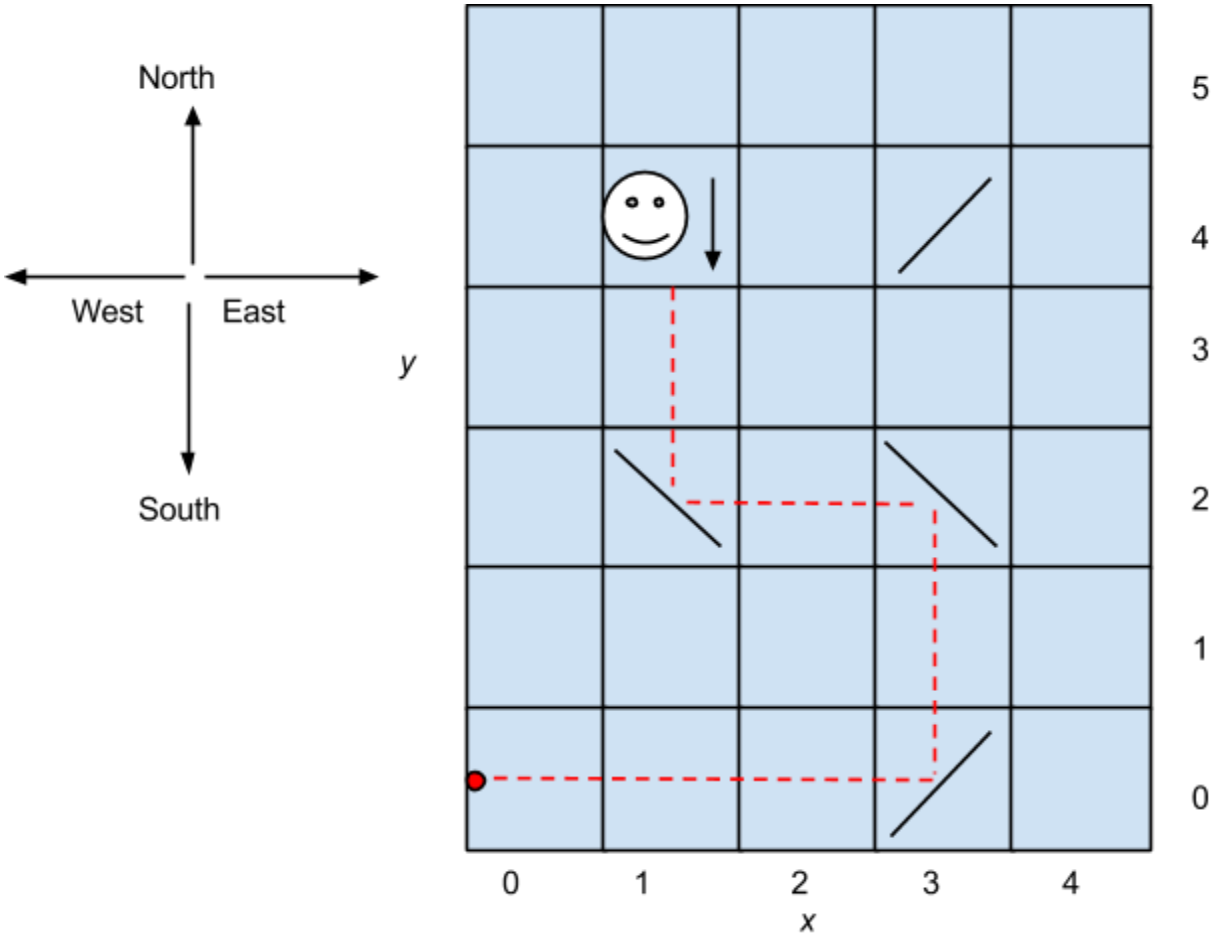
The output file must always be in the following format:

```
9    # number of squares traversed, return -1 if wall is never hit
0 0  # coordinate of final square (unnecessary if wall is never hit)
```

The solution data must be written to the path of the output file (second argument when invoking the command line utility).

Visualization

The sample input is visualized below and will correctly provide the sample output above.



Design

Please consider the following criteria for your implementation:

- Object Oriented: your code should be modular, flexible, and reusable
- Well-Tested: follow your best test-driven practices, and implement a test suite in your framework of choice
- Extensible: it should be easy for both junior and senior engineers to improve and add features to your code.