Maze Runner

Given a maze with walls and open spaces, you must determine if a robot can travel from the starting location to the finish! *Thanks to Dr. Blaheta for the idea!*

Extra Credit:

This assignment is purely optional and will count as an additional free homework score worth 100 points. For example, if you have received a 100 on all homework assignments, then you could possibly earn 1000 out of 900 points for your homework grade in the class.

Introduction:

The key idea is that you have to help your robot determine if it is possible to make its way through a maze. The robot will be given a particular starting location in the maze and a particular square to finish at. The robot may only travel left, right, up, and down (no diagonal moves!). The maze is represented as a text file with the following numeric values:

Value	Meaning
0	Open square your robot can travel on
1	Wall that blocks the robot's path
8	Starting location of the robot
9	Finish location for the robot

The maze will always be a square grid (e.g. 10 by 10, 12 by 12, 20 by 20, etc...). For example, here is a maze that is **12** by **12** in size:

```
1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 0 0 0 0 0 1 0 1
1 0 1 0 1 1 1 1 1 1 0 1
1 0 1 0 0 0 0 0 0 0 0 1
1 0 1 0 1 1 1 1 1 1 0 1
1 0 1 0 0 0 0 1 0 0 0 1
1 0 1 1 1 0 9 1 0 1 0 1
1 0 0 0 1 1 1 1 0 1 0 1
1 0 1 0 1 0 0 1 0 1 0 1
1 0 1 0 1 0 1 1 0 1 0 1
1 8 1 0 0 0 0 0 0 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1
```

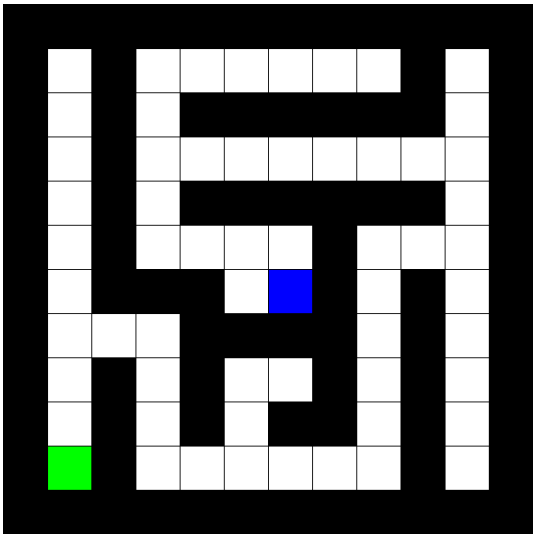

Figure 1: Maze visualization.

Key steps to solution:

This assignment is broken into two parts, each worth **50** extra credit points! The first part is simply drawing the maze, and the second part analyzes the maze for a possible solution. You are welcome to do both parts, or only one!

1. Part 1: Visualize the maze
   (a) Goal: Create and save an image of the maze (**use imwrite to save the image!**
   (b) Read the file **maze.txt** to see the shape of the maze. Remember, you don't know the size of the maze until you process the file!
   (c) Use the following color codes:
      - Walls = Black square
      - Open space = White square with black outline
      - Start location = Green square with black outline
      - Finish location = Blue square with black outline
2. Part 2: Analyze the maze to see if it has a solution (You **do not** need to find the solution!)
   (a) Goal: Simply display **Maze has a solution!** or **No solution possible...**
      - Begin at the starting location of the maze and check if the neighboring spaces are open (so check left, right, up, down)
      - If you find any open spaces, check the neighbors of those spaces. **Keep repeating this process for the neighbors of the neighbors!**
      - Stop the the algorithm when you have found the finish location or there are no more spaces to check!

Sample Output:

There are none! You are given one example maze to test with, but you should make your own mazes and test those.

What to turn in:

First, make sure your script executes without error to receive full credit.

1. Name your source file (.m) with your **UF User Name** and `_HW10`.
   - Example: `ijh3_HW10.m`
2. Upload your .m file to Canvas through the assignments page for Homework 10. Please double check the name is correct before submitting!