HW 2 Questions

**Question 2:**

Given the algorithm, main function, and maze shown at the end of problem 1, what are the first 12 (r,c) coordinates popped off the stack by the algorithm?

Pop:

(3,4), (3,3), (3,5), (2,5),

(1,5), (1,6), (1,7), (1,8),

(2,8), (3,6), (4,4), (5,4)

**Question 4:**

1. **Given the same main function and maze as are shown at the end of problem 1, what are the first 12 (r,c) coordinates popped from the queue in your queue-based algorithm?**

Pop:

(3,4), (3,3), (3,5), (2,5),

(1,5), (1,6), (1,7), (1,8),

(2,8), (3,6), (4,4), (5,4)

1. **How do the two algorithms differ from each other? (Hint: how and why do they visit cells in the maze in a different order?)**

The stack-based algorithm performs a depth-first search which branches out in one direction before moving to another. This happens because the stack algorithm uses the Last-In-First-Out approach. This results in the program concentrating down one pathway until its end. For this assignment, we search South -> East -> North -> West. So, the program will always start at a west coordinate if available, otherwise north, otherwise east, otherwise south, otherwise the path is at its end and we need to go down another path.

The queue-based algorithm uses a breadth-first search which explores the maze evenly. Opposite of the stack-based algorithm. This is because unlike the stack, the queue follows the First-In-First-Out approach which results in evenly traveling down all available paths. From the first coordinate, we observe all points around it to see what paths are available. Then the algorithm will travel down each of those paths evenly until the end(i.e. jumps from one path to another).