CS32: Homework #2

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?

**(row, col):**

1. (6, 4)
2. (6, 3)
3. (6, 5)
4. (7, 5)
5. (8, 5)
6. (8, 6)
7. (8, 7)
8. (8, 8)
9. (7, 8)
10. (6, 6)
11. (5, 4)
12. (4, 4)

4. 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?

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

Because the first algorithm was implemented using a stack, and the second used a queue, the behavior differs.

A stack is a LIFO structure, meaning that the last element pushed will be the first element to be popped if called to do so. In our algorithm’s case, the stack implementation will consider the top of the stack, and continue along the path until it reaches the passed endpoints or its stuck. To do so, the most recent coordinate that was pushed to be the first popped, thus popping the above sequence of coordinates in the maze visited in that order.

A queue, however, is a FIFO structure, such that the first element enqueued will be the first to be dequeued if called to do so. The queue implementation will consider the oldest enqueued value’s surroundings (in the order N, E, S, W) and follow that path. Therefore, the queue-based algorithm aligns with FIFO behavior, such that it causes the coordinates to the left to be dequeued in this manner.

1. (6, 4)
2. (5, 4)
3. (6, 5)
4. (6, 3)
5. (4, 4)
6. (6, 6)
7. (7, 5)
8. (3, 4)
9. (4, 5)
10. (8, 5)
11. (2, 4)
12. (4, 6)