1. 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?  
   For this problem, you'll turn in either a Word document named hw.docx or a text file named hw.txt, that has your answer to this problem (and problem 4).
2. (6, 5)
3. (6, 6)
4. (6, 4)
5. (7, 4)
6. (8, 4)
7. (8, 3)
8. (8, 2)
9. (8, 1)
10. (7, 1)
11. (6, 3)
12. (5, 5)
13. (4, 5)
14. 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?)  
    For this problem, you'll turn in either a Word document named hw.docx or a text file named hw.txt, that has your answer to this problem (and problem 2).
15. (6, 5)
16. (5, 5)
17. (6, 4)
18. (6, 6)
19. (4, 5)
20. (6, 3)
21. (7, 4)
22. (3, 5)
23. (4, 4)
24. (8, 4)
25. (2, 5)
26. (4, 3)

A stack is considered a last in first out algorithm, while a queue is a first in first out algorithm. This means that when you pop a Coord out of the stack, you pop the last coordinate that went in, while when you pop a Coord out of the queue, you pop the first coord that went in. This means that with a stack, it would add east, south, west, and then north for the first coordinate (whichever directions it could), and then it would go backwards when you pop. This means you would pop the East coordinate first, and then it would add east, south, west, north for the East coordinate, and you would pop the East of the original East coordinate next and so on. On the other hand, for the queue, it doesn’t matter what you add in the second round of the while loop, it will still go from the front of the added coordinates, meaning north of the first Coord would be dealt with and then West, South, and East in that order, and then North, West, South, East of the original north Coord, and so on.