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

(4,3)

(3,3)

(5,3)

(5,2)

(5,1)

(6,1)

(7,1)

(8,1)

(8,2)

(6,3)

(4,4)

(4,5)

4. The first 12 coords from the queue are:

(4,3)

(4,4)

(5,3)

(3,3)

(4,5)

(6,3)

(5,2)

(4,6)

(5,5)

(5,1)

(4,7)

(6,5)

The algorithm in mazestack.cpp uses depth first search; the algorithm in mazequeue.cpp uses breadth first search.

To clarify, with a stack, the algorithm searches down a path to the very end before backtracking to a point where it can continue searching all the way down to the end of another path, repeating until it finds whether or not a maze can be completed. This is because stack is first in, last out – the items will keep being added to the top, popped off, and search will continue from that last item looked at.

On the other hand, with a queue, the algorithm searches like a ripple; it spreads out across all the different paths little by little until it finds whether the path can be found or not. This is because queues are first in first out. When different coordinates to check are added to a queue from multiple different directions, they will come out in the order they were put in. So a search could go something like east, south, and west from a point, and then search south and north from a different point that had gone down a different path.