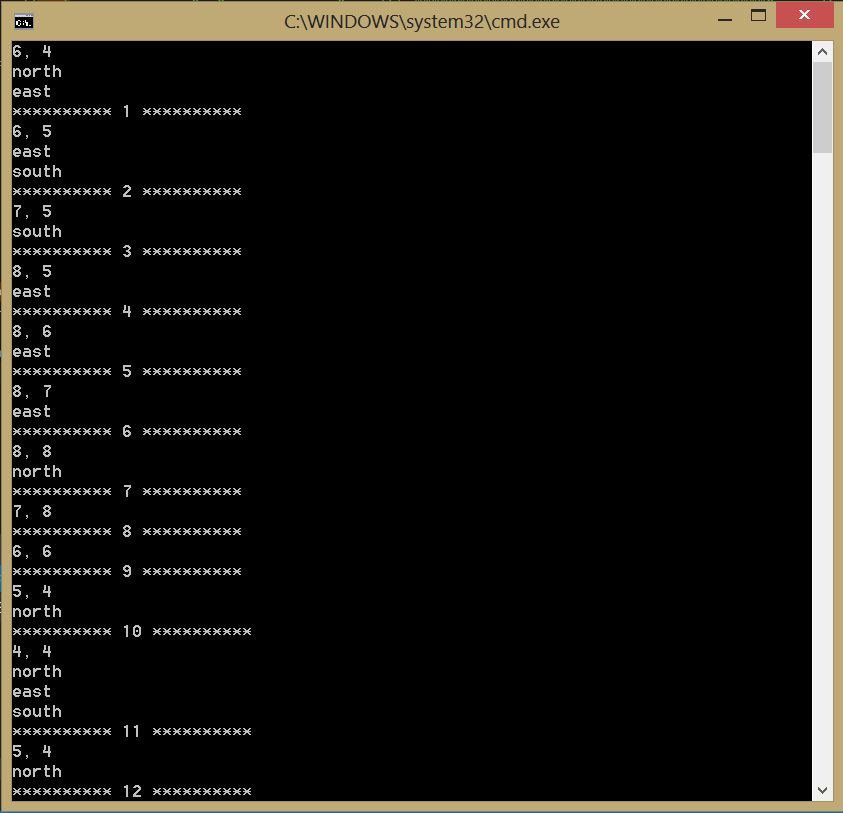
**Homework 2**

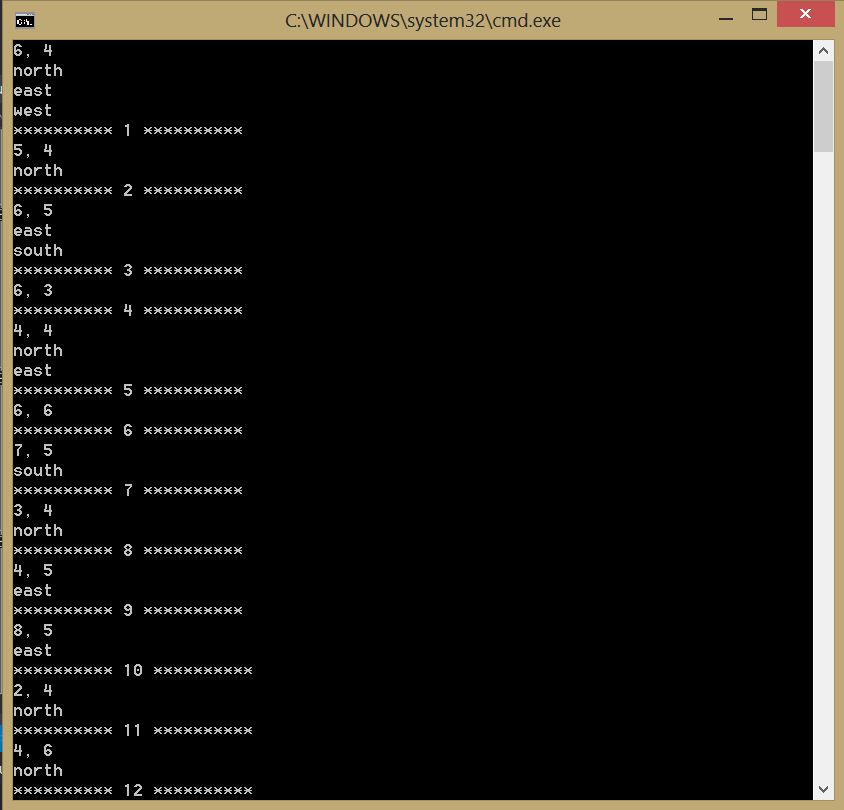
2) With the given algorithm and the main function given, the first 12 coordinates ‘popped’, i.e. marked with ‘0’ here, are:



The reason for such behavior is explained in (4).

4) STACK: As we move through the loop, we push the coordinates of every walkway around the current position into the stack. We prioritize the direction to start as North->East->South->West, and the coordinates of every ‘fork’ is pushed into the stack. If the north direction takes us to a dead end, we keep popping the top item until we reach the last fork where we decided to follow the priority path (i.e. N->E->S->W) and re-mark the already travelled path with ‘0’. This way we can explore all possible routes and check if a path to the end exists.

QUEUE: As we move through the loop, we queue the coordinates of every walkway around the current position into the queue. We prioritize the direction to start as North->East->South->West, and the coordinated of every ‘fork’ is queued into the queue. However, unlike the stack application, the queue data type ‘pops’ items from the front of the list. This is why at every fork, the viability of a path is tested in the reverse of the priority order (i.e. W->S->E->N). Just like in the stack application, if any given path leads to a dead end, the coordinate of the last ‘fork’ is de-queued, and we continue testing from the fork before it. Consequently, the first 12 coordinates de-queued are:



NOTE: In both cases, the directions denote the possible locations you can move from the coordinates of the previous ‘forks’. As we proceed, the possible locations grow, and then decrease as we continue to find and check the viability of the paths respectively. This is also why it might appear the program ‘jumps’ from one location to another; what’s really happening is that if the person reaches a dead end, instead of traveling back from the dead-end to the last fork, it just starts exploring from the other direction at the last fork. The printed coordinates only depict the coordinates that are marked as “already visited”.