CSE 674 HW4 Due: 11:59PM Friday (Oct. 11)
Objectives: Practice BFS and DFS, and apply them to solving mazes.
You are required to implement as described in class.

Generate one meaningful 20 by 20 maze, and include it in the end of your code.

Your program will read in a maze from a file, maze.txt.

Search the maze and find the path from start to finish using both depth first algorithm and breadth first algorithm. Note that both DFS and BFS are implemented as functions. At each cell, if multiple neighbor cells can be pushed to stack or queue, follow the sequence: S-E-N-W. You can use stack and queue from C++ STL.

```
void DFS(vector<cell> &maze, int n, int start, int end);//end is target cell
void BFS(vector<cell> &maze, int n, int start, int end);
```

Output should looks like the following:

```
Using DFS:
0 -> 1 -> 5 -> 7 -> …  or No solution.
Using BFS:
0 -> 4 -> … or No solution.
```

Bonus: Print out the maze and its solution two-dimensionally.

```
class cell {
public:
        int id;
        int wall;
        bool visited;
        bool in_structure;
        int from;
        //design your constructor
};
```

Format for maze.txt:
16 //number of cells
0//starting cell
15//target cell
8 4 5 6 10 9 6 10 8 6 11 10 11 9 7 10//cell walls