Department of Electrical and Computer Engineering
EECE 2560: Fundamentals of Engineering Algorithms

Project #4
Part b

1. Write a recursive function `findPathDFSRecursive` that looks for a path from the start cell to the goal cell using depth-first search.
2. Write a function `findPathDFSStack` that that looks for a path from the start cell to the goal cell using depth-first search, but using a stack and not recursion.
3. Write a function `findShortestPathDFS` that finds a shortest path from the start cell to the goal cell using depth-first search.
4. Write a function `findShortestPathBFS` that finds a shortest path from the start cell to the goal cell using breadth-first search.

The code you submit should read the name of maze file from the keyboard, and then apply each functions to that maze, one after the other.

If no path from the start to the goal exists, the program should print, `No path exists`.

If a solution exists, the solver should simulate the solution to each maze by calling the `maze::print()` function after each move.