**Programming Project Report**

Name: Logan Campbell

Date: 6/7/17

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

The goal for this programming assignment was to create a program that identified characters for a maze and show the solution to such maze. The inputs would have to be the input read file that was created and used in the program. The output would show the loaded maze and the solution to the maze. The only error handling would have to be the file name and that was to open the file if it existed.

**Design:**

The design of the program started out with function that are listed:

Load maze which opened the file of the maze. Print maze which output the board of the maze. Entry point identified what qualified as an entry point. Find entry point would use the dimensions and look for such point. Solve used a board that was unknown to the program and would use find entry to place the setup for the solution board. Solve helper is where the algorithm is. Solve helper used a base case that identified the end point of the maze and then would use itself recursively to identify other characters that were spaces. From there it would identify what was a valid move and what can be placed in the solution board.

**Implementation:**

Using the given pdf to identify the structure of the program. Create:

Load maze, print maze, entry point, find entry point, solve, and solve helper.

Load maze was used to take the txt file and read it.

Print maze was used to output the maze

Entry point was used as a conditional function and returns the answer.

Find entry point was used to find the entry point within the array. It would also use entry point as a condition.

Solve: creates find the solution

Solve helper: would find the solution within the whole board places in solve.

**Testing:**

Testing the program was fairly easy.

Load Maze & Print Maze: I used the maze from the pdf and see if it would print out.

Entry Point & Find Entry Point: I use the solve/load/print function to identify if the first space and last space (when I didn’t use the first path on the outside of the maze) of the entry point and cout to see if it was correct.

Solve Helper: was the most difficult. Assuming that everything worked this function would use itself to move through the maze. Knowing how to loop through each section would put the path in the solution in which... solve would contain the answer.

Solve: by seeing if the solution was correct.

**Conclusions:**

Everything went smoothly except the solve helper it was a little confusing. I haven’t tried the extra bonus points because I really didn’t want to create a catastrophe but I’ll try. Regardless, I think I would have done everything the same except finding the end entry point. It seemed that I would have been easier to find the end of the maze if you had it in solve and had solve helper have a condition to stop recurring if it reach the end.

It took about 4 days on and off.