Marawan El Sharkawi

CSCI 301-Section 1

#cs301175

Project 3

**Design Document**

**Introduction**

Recursion is a technique in programming which allows a function to call itself. It is one of the ways to make repeated computations. A recursive function passes a smaller version of the problem to a new function of itself then it uses the results returned from all the functions it has called to determine the solution of the main problem. This program reads a maze from a file displays it then find paths from its start to its end using recursion.

**Data Structures**

The program uses three data structures:

* Two integers variables called “r” and “c”, these two integers hold the numbers of rows and columns of the maze respectively being read from the file contains the maze.
* Three char variables, one called “lineEnd” to hold the character at the end of the line to start reading a new line. The second called “space “which the program uses to print a blank space whenever there is no walls in the maze. And the third one called “path” the program uses it to print a dot whenever there is a path that solves the maze.
* And a two-dimensional array called “maze”, this array holds very character read from the file, rows of the maze in one of its dimensions and columns in the other dimension. A program constant sets the maximum value of the numbers of rows and columns of the maze the program can read and find a path through.

**Functions**

The program uses four functions:

* openFile() , it takes ifstream which is a stream class to read from the file contains the maze as a parameter and its being passed by reference. The function asks the user to input the name of the file contains the maze to open it, if the name of the file does not exist or the user entered a false file name the function keep asking the user to enter the name of the file until the users enters a correct file name.
* readFile(), it takes ifstream parameter, maze two dimensional array, and two integer variables passed by reference to refer to the number of rows and columns of the maze. The function reads the number of rows and columns of the maze from the opened file, saves the characters of each row and column of the maze into the two-dimensional array.
* display(), it takes maze two dimensional array, and two integer variables to refer to the number of rows and columns of the maze, the functions prints out the characters saved into the two dimensional array which are the maze read from the file.
* pathFinder(), it takes maze two dimensional array, two integer variables to refer to the dimensions of the start position of the maze and two integer variables to refer to the number of rows and columns of the maze. The function finds all the possible paths from the start position of the maze till the end position, prints a space whenever there are no walls and prints a dot wherever there is a path that will lead to the end of the maze, the function find the paths using recursion technique.

**The main program**

The program creates an ifstream variable for the file the user will input, two integer variables one to hold the number of the rows of the maze and the other to hold the number of columns of the maze, and a maze two-dimensional array. The program then calls the openFile() function and passes the ifstream variable as its parameters then calls readFile() function and passes the ifstream variable, the maze variable and the two integer variables as its parameters. Next, the program closes the file contains the maze after the readFile() function is done with its work, after that it calls display() function and passes the maze variable and the two integer variables as its parameters. The final step is to assign the dimension of the start position of the maze through the two-dimensional array then to call the pathFinder() function which will find the paths through the maze and print out the maze with the paths found.