

Hacettepe University Computer Engineering Department



Name: Habil

Surname: Ganbarli

Student ID: 21501017

Assignment number: 1

Problem: Finding path in Maze using C programming language

AIM

Aim of this assignment is to gain practice on Data Structures and Learn C language by means of implementing **Recursion** in order to solve Maze pathfinding problem.

Problem

Since the data is one of the important subjects of this assignment, first one should concentrate on the shape of the maze. The maze is in a text file as a form of matrix. The size of the matrix can change and that is why the program should be dynamic for the different sized matrixes in that file. The character “0” is considered as passing point and “1” is considered as barrier in this assignment. However, the matrix also contains some letters which are both capital and low case. The low case letters are the keys for the corresponding (same type) capital letters which are doors. We are allowed to use four directions which are North, South, West and East.

The main goal is to start from “S” then pass from keys and go to doors. Moreover, our last aim is to exit from E. The path should be logged according to its corresponding Direction and final should be written to another path text file.

Example Input:

S	0	0	1	0	0	1
1	1	0	1	0	1	0
0	0	0	1	0	1	0
a	0	1	1	1	0	0
0	0	0	A	0	0	0
1	1	0	1	1	1	0
0	b	0	0	0	B	E

Example Output:

Start	E	E	S	S	W	W	S	a	S	E	E	E	A	E	E	E	S	S	Exit
-------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

SOLUTION

First concentration point is to how to treat matrix in order to solve the maze. In order to solve this problem, we have to put the matrix into array and treat them as **x, y** coordinates and go through them in order to find the path. Since we have four directions (N, S, E, W) to go, the displacement on every point should be according to that. Therefore, North, South, East, West would be **(x-1, y), (x+1, y), (x, y+1) and (x, y-1)** respectively. Additionally, since not every point is appropriate to pass we have to check them beforehand.

As a main data structure for this assignment, I have used 3 2D arrays in this assignment. The first one just contains the data which is in the text file. The second one is to mark the tried coordinates in order not to pass from them again. The third one contains the solution route.

First step is to find the size of matrix through helper **numOfLines()** function from text file. Secondly, I used 2 other main functions:

isSafe()

This function checks the coordinate if it is “0” or “1” and also makes sure that the coordinate is not out of the matrix

recursiveMaze()

This function is the key function to solve the problem which functions recursively in 4 directions, marks the tried parts and also prints the path.

The Program’s functionality

Despite the fact that the program should collect keys and pass from doors, my solution only finds the shortest path from start point to exit. It ignores the keys and doors whether it passes from the according to input.

What I have Learnt

- 1. The C language general syntax**
- 2. How to use arrays effectively in C**
- 3. The use of complex recursive functions**
- 4. File I/O in C**
- 5. Methods to solve Maze problems**

