Hacettepe University

Computer Science and Engineering Department

Name and Surname	***: Tolgahan DİKMEN***
ID Number	***: 21327929***
Course	***: BBM203 Programming Lab.***
Experiment	***: Assignment 1***
Due Date	***: 22.10.2017***
Advisor	***: R.A Gültekin IŞIK***
E-mail	***: tolgahandikmen95@gmail.com***

This report belongs to explaining code of maze problem. The problem consists of two different parts. First part starts creating a maze which is given by user via .txt file, then program need to find an exit way. Second part comprise that organize the valid way to exit.

The aim of this assignment is operations in two dimensional array dynamically and improve logic of recursion. It must use to dynamic memory allocation because the maze size is changeable according to the input file.

Variables and array which has been used:

- **char argv[1]:** gets name of the input file.
- **int starting_point:** holds the starting point, it is an integer number because starting point always be on the first row. Actually real point of starting is (0, starting_point).
- **char path[ ]:** will be holds directions of the exit way.
- **int key[ ]:** will be holds the keys of doors.
- **int path_counter:** holds to the last position of path[ ] array.
- **int key_counter:** holds to the last position of key[ ] array.
- **int exit:** it is a controller value, if it is '0', exit could not find, otherwise found the way to exit.
- **int **maze[ ][ ]:** it is two dimensional dynamic array of the maze like a matrix.

The maze and the key array have int type because the content of the file which has read, are character. But they are held with integer values as for that ASCII numbers.

Firstly, program reads only first row of input file for size of matrix which will be created. Then two dimensional array will be created. After maze is ready to operations, continues with searching 'S' letter in first row for starting point. After found starting point then calls the recursive function.

move (int maze, int matrix_size, int i, int j, char* path, int path_counter, int* key, int key_counter, int* exit_flag, FILE *outputFile)**

- **maze:** to be constructed maze
- **matrix_size:** will be used in loop statements or comparisons
- **(0,start_point):** starting point of the maze
- **exit_flag:** it is a pointer because if it changes once, other recursive move functions need to be influenced
- **outputFile:** will be written file

move( ) Function

The function takes a point which consist the values of latitude and longitude; path and key arrays; maze matrix and other arguments what it needs to know. Main logic is moving step by step. Operations will be affected latitude and longitude which comes to the function. Controls some specific points then tries to go to the neighbour cells if it could be gone.

Process is going to continue until reaching to 'E' Exit or all maze is checked also could not find any exit way.

The function starts if statement outmost. It checks *exit_flag pointer. It can go to inner if exit_flag is not 0. The function comes to an end if the cell 'E' could find. The path array is written to output file.

After all processes, memory which is taken for maze matrix is released.

This assignment is very educational for recursion and dynamic memory management. Also pointers need to be used for structure. In a given sense this assignment comes up with the time and difficulties in a parallel way.