HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING
BBM203

Prepared by Burak YILMAZ
Number : 21627868
E-Mail : burak040898@gmail.com
Subject : Assignment 1

MAIN GOAL

In this experiment, our aim is find the hidden treasure within a treasure map designed as a matrix.

PROBLEM

In this problem we need to write a C code that the map and key data must be read from the file. Afterwards, the key must be slide on the map depending on the rules. The result of the multiplication of the map matrix and the key matrix will give an idea of the place of the treasure.While solving this problem we need to use recursion and 2D arrays(not static).Also we have to use matrix structure and dynamic memory allocation.

SOLUTION

I used a matrix structure that include a row,column and double pointer for keep the map_data.I allocate a dynamic memory inside that map. I read the data from the txt files and successfully keep them.So when I come main part of this Assignment(recursive part)I first decided the base case for this assignment when the calculation ends if the result is “0” then recursive code should stop.So I encoded my C code considering the rules that mentioned on the pdf.Finally I deleted the memory that I allocated for the 2D array(using double pointer and malloc).

ALGORITHM

Firstly, I read two input files(mapmatrix and keymatrix). After reading I split the matrix argument(for example 18x18) to determine the boundaries of the matrix and considering these values I allocated the memory by using dynamic memory allocation.I continued with storing the values of the map inside that memory.So after these operations the only thing left was the recursive part.Considering the rules of the pdf I encoded that part and done with the assignment.Finally I used free method for releasing the memory that I allocated before for storing the data.

FUNCTIONS/METHODS/STRUCTS

```
typedef struct matrix{
int rows;
int columns;
int **map;
} matrix;
```

I used this struct for storing the matrix.

```
void Matrix_reader(char *,matrix*);
```

-This function reads the maps from the given input as an argument.Also at the same time It allocates the memory based on the matrix rows and columns.Finally after allocating it keeps the data inside that allocated memory.It takes two arguments.One of them is character pointer the other one is matrix structure pointer.

```
void Free_Allocation(matrix*,matrix*);
```

-This function release the allocated memory before to prevent the memory leak..It takes two arguments as a matrix type(matrix structure pointer). Releasing the double pointer.

```
void Matrixes_Boundaries(char *,char*,matrix*,matrix*);
```

-It determine the map_matrix boundaries and key_matrix boundaries considering the given inputs as an argument.It takes 4 arguments.Two of them is character pointer and two of them is matrix structure pointer.

```
void Treasure_Searcher(matrix*,matrix*,int,int,int,int,int,FILE*)
```

-This function works recursively.Base case for the recursive funciton if the remainder from the calculation equals to 0 it stops otherwise calls itself recursively.Also it checks the boundaries of the map if the boundaries is not enough for sliding the key map on the map_matrix it changes the direction of the sliding and continues from like that.For each recursive call it prints the center of matrix and the result to the output file.It takes 8 arguments.Five of them is integer and two of them is matrix structure pointer and one of them is file pointer.Integer arguments for the boundaries,file pointer is for writing the center of matrix and the sum of the calculation to the output file.Type of matrix pointers for accesing the data inside them.