HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING

BBM 203 PROGRAMMING LAB. ASSIGNMENT 1

Name: Ahmetcan
Surname: Gürel
Number: 21727297
Advisor: R.A. Alaettin Uçan
Subject: Data Structures and Algorithms

Problem Definition: In this experiment, it is aimed to find the hidden treasure within a treasure map designed as a matrix.

For this purpose, the map and key data must be read from the file. Afterwards, the key must be moved 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.

We should use matrix structure to implement all these operations in the C programming language. To store the matrices, we use dynamic memory allocation.

Functions and Algorithm:

- printResult function

I used the printResult method for writing the output to the output file.

-newMatrix function

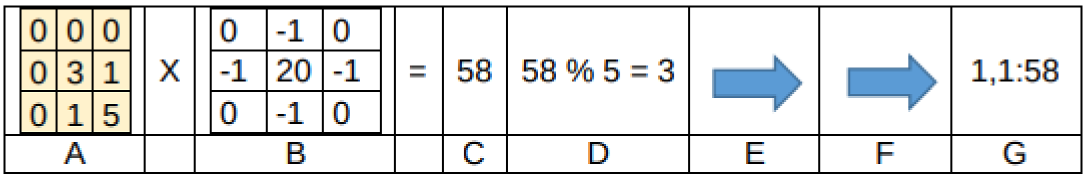
Taking the map and the comparison matrix from argument and reproducing the matrix for another comparison. I use this method in all cases except if the final result is “0”.

-findTreasure function

Taking the map,keymap,counter,tempr,tempc and the comparison matrix from argument. I define an integer “index” for finding the middle index of the results. To find the index I do the following mathematical operation:

    index = counter/2;

Counter is the key matrices size so I divide the counter with 2 to find the middle index.
After that I do the multiplication operation in a for loop between the matrix and the key map. Example of the multiplication :



$\begin{matrix} 0 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 5 \end{matrix}$	X	$\begin{matrix} 0 & -1 & 0 \\ -1 & 20 & -1 \\ 0 & -1 & 0 \end{matrix}$	=	58	58 % 5 = 3	➡	➡	1,1:58
A		B		C	D	E	F	G

Then I define an integer “finalresult” for finding the final result of mod operation. I do the following mathematical operation:

```
finalresult = multiplication % 5;
```

And then the if-else if statements begin. I have 5 if-else if statements for the result of the mod operation. If the result is:

0: Found the treasure
1: Go up
2: Go down
3: Go right
4: Go left

Inside of these statements except “0” statement I have another if-else statements to look for the index exceed. It looks for the indexes if they all are “0” it must be the boundries so the matrix goes to the opposite direction. Then reproducing the matrix by using the “newMatrix” function. Finally going to the “findTreasure” function recursively. If the result is “0” the recursive is finished.

-main function

In main function I define some integer values named “keysize, keymatrixsize, tempr, tempc”. Afterwards using the “sscanf” method I take the map size from the command line. Then I take key size from the command line and using “atoi” method I turn string value to integer value. After that I create 3 file pointer for the input files and output file. Then I reserving RAM for the map matrix, key matrix and the comparison matrix and add the values of them using for loop. Lastly I call the “findTreasue” function and close the files and free the memories from the RAM.

Methods

- sscanf ,fscanf >>> Reads formatted input from a string.

- atoi >>> Converting a string value to an integer value.

- fopen >>> Opens a file.

- fclose >>> Closes a file.

- fprintf >>> Writes to a file.

- free >>> Frees space from the RAM.