Hacettepe University
Computer Science and Engineering Department

Name and Surname	**:** Necati Berk ÖZGÜR
Identity Number	**:** 21785229
Course	**:** BBM203
Experiment	**:** Programming Assignment I
Subject	**:** Find Treasure
Date Due	**:** 04/11/2018
Advisors	**:** R. A. Alaettin UÇAN
e-mail	**:** b21785229@cs.hacettepe.edu.tr

2. Software Using Documentation

2.1. Software Usage

By this software, user is able to play a treasure finding game by given key and map.

2.2. Provided Possibilities

-Built-in boundary checking.(i.e Changes direction when reached an edge.)
-Prompt the center point where player stands at given time.

2.3 Error Messages

This program handles no errors.

3.Software Design Notes
3.1. Description of the program
3.1.1. Problem

A treasure is hidden in given map.

3.1.2.Solution

Player moves given key over map to find treasure.

3.2. System Chart

INPUT	PROGRAMS	OUTPUT
Treasure Map	*findtreasure*	*Output is prompted to file given by player.*
Key		
Map Size		
Key Size		

3.3. Main Data Structures

“Multidimensional dynamic array” is the key data structure in this experiment. Besides, **int** and **string** types are also used.

3.4. Algorithm

1. Make initialization:
   1.1. Open input file(s).
        With built in C function **fopen()**.
   1.2 Generate proper Key and Map matrix according to arguments given.
        createmap(FILE*, int, int) creates proper maps according to given sizes(integer parameters) and given file(FILE type parameter)

2. Gameplay (**findtreasure(FILE*, int**, int**, int, int, int, int, int)**):

findtreasure(); function takes output file, map, key, number of rows and columns, and initial start positions. And seeks for treasure **recursively**.

2.1 Calculate initial(position: 0,0) Map-Key value.(Multiply every element in key with corresponding element in map at given position, then add everything up. After then, modulo 5 the result.)

2.2 Move key according to pre-defined directions until the treasure hidden is found.

0 : Found Treasure
1 : Up
2 : Down
3 : Right
4 : Left

3. Print center positions to out file, stop when treasure found.

fprintf();

4. Close files and free pointers allocated.

Fclose(), free().

3.5. Specaial Design Properties

In this design, “***malloc***” function is mainly used. Also another key feature **find_treasure** function is designed as a recursive function.

By the ***malloc*** function which is a built in C function, map and key can be stored in variable sized arrays,
By the recursive **find_treasure** function, code became more readable and clean.

4. Software Testing Notes

4.1 Bugs and Software Reliability

By the boundary checking, and auto-redirecting, player is not going to face with such errors like “array out of index”.
But program is not protected against map-key matches which may cause infinite loops. (This error can be handled with a limiter parameter in **find_treasure** function.)

4.2 Software Extendibility and Upgradability

By modular construction(functions, little use of main function, etc.) of C language, the code is easily upgradable.

REFERENCES

Pressman, 1987 (Supplied by ftp of Hacettepe University Dep. of Computer Science and Engineering)
The C Programming Language, Ritchie – Kernighan, 1988, Prentice Hall
Fundamentals of Data Structures in C, Horowitz – Sahni, 2008, Silicon Press