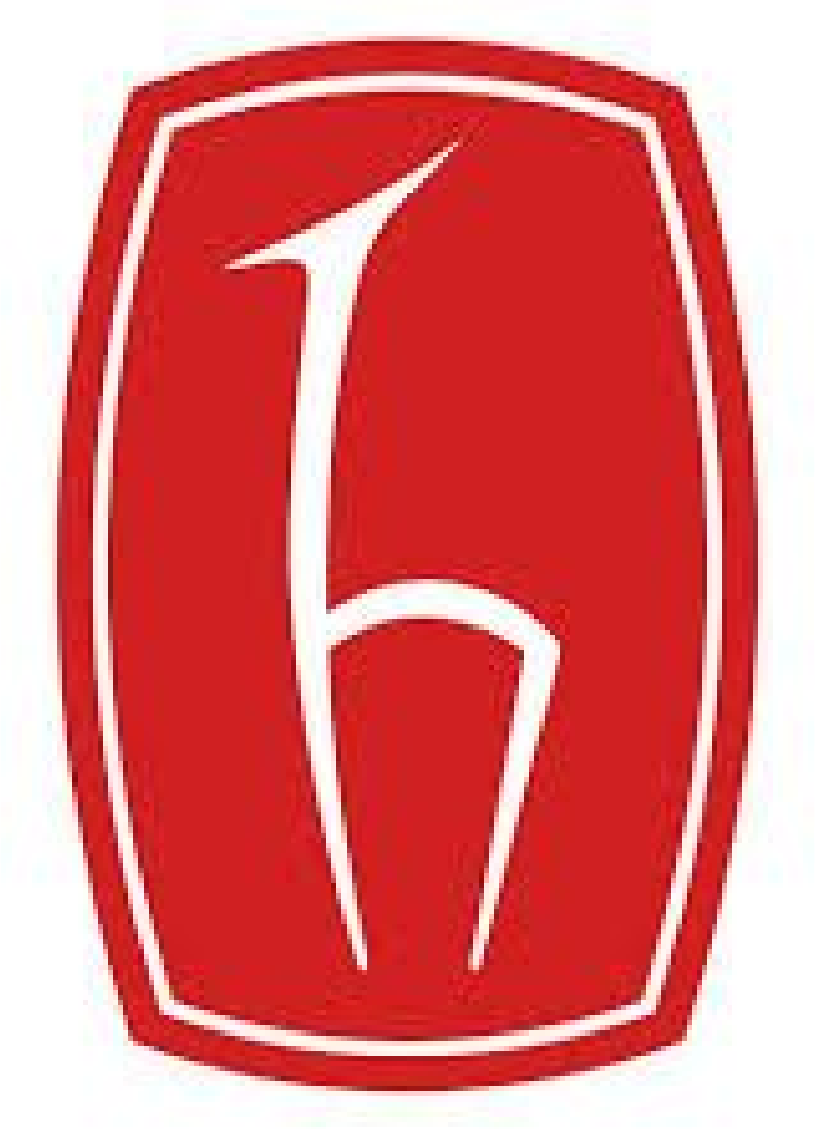

ASSIGNMENT 1

Subject : Data Structures and Algorithms

Advisor : R.A Alaettin Uçan

Programming Language: C

Due Date: 13.11.2019 23:59:59

Name:İsmail ATEŞ

Number:21626953

INTRODUCTION / AIM:

In this assignment, we are expected to gain knowledge on C language including read-write files, arrays, matrices, recursion and dynamic memory allocation. Prior knowledge on basic C syntax is assumed.

PROBLEM DEFINITION:

Our task is, storing vector and matrix files on program. It's creating changes with, reading file given name of folder or with create inputs. Then we are perform some mathematical operations and other things on matrices and vectors. We need to dynamic memory allocation and dynamic program.

STRUCT THAT I USED:

```
struct file {
    char *name;
    int **array;
    int row;
    int column;
};
```

File struct stores:
- vector/matrix name
-vector/matrix 1D 2D array
-vector/matrix row count
-vector-matrix column count

FUNCTIONS THAT I USED:

void veczeros(**struct** file *filearray, **int** *file_count, **char** *name, **int** length, FILE *pFile)

Function checks name in filearray if it's in it, function prints error. Otherwise it creates vector named 'name' with length 'length' and fill's elements zero.

void matzeros(**struct** file *filearray, **int** *file_count, **char** *name, **int** column, **int** row, FILE *pFile)

Function checks name in filearray if it's in it, function prints error. Otherwise it creates matrix named 'name' with column length 'column' and row length 'row' .Then fill's elements zero.

void vecread(**struct** file *filearray, **char** *v[], **int** *file_count, **char** *name,FILE *pFile) {

This function takes vector name with file extention.Concatenate folder of matrix/vectors name and vector file's name and open it. And prints elements of vector.

void matread(**struct** file *filearray, **char** *v[], **int** *file_count, **char** *name, FILE *pFile)

This function takes matrix name with file extention.Concatenate folder of matrix/vectors name and vector file's name and open it. And prints elements of vector.

void vecstack(**struct** file *filearray, **int** *file_count, **char** *vec1, **char** *vec2, **char** *direction, **char** *matName, FILE *pFile)

This function checks matName in filearray for is it created before or not. Then it checks vector1 and vector2 length for are they appropriate for vecstack.If it's not appropriate, prints error. Otherwise function creates new matrix with elements of vector1 and vector2.

void matstack(**struct** file *filearray, **int** *file_count, **char** *matrix1, **char** *matrix2, **char** *direction, **int** **copy, FILE *pFile)

This function check matrix1 and matrix2's row and columns for are they appropriate for matstack. İf it is not appropriate, prints error . Other wise matrix1 resized and fill with new elements.

void mvstack(**struct** file *filearray, **int** *file_count, **char** *matName, **char** *vecName, **char** *direction, **int** **copy, FILE *pFile)

This function check matrix and vector row and columns for are they appropriate for mvtstack. İf it is not appropriate, prints error . Other wise matrix1 resized and fill with new elements.

void pad(**struct** file *filearray, **int** *file_count, **char** *name, **int** column, **int** row, **char** *mode, **int** **copy, FILE *pFile)

This function resize matrix and fill elements with mode. If mode maximum, for new row elements, it finds maximum element of this row and fill the new rows with this maximum element. The same goes for column. If mode minimum steps are same but new elements were fills with minimum values.

voidpadval(**struct** file *filearray, **int** *file_count, **char** *name, **int** column, **int** row, **int** value, **int** **copy, FILE *pFile)

This function resize matrix and fill elements with mode. New elements of matrix fills with given value.

void vecslice(**struct** file *filearray, **int** *file_count, **char** *name, **int** start, **int** stop, **char** *newName, FILE *pFile)

This function checks if new vector already exist in file array. If it is it's prints error. Otherwise it slice the vector and create new vector.

void matslice(**struct** file *filearray, **int** *file_count, **char** *name, **int** y1, **int** y2, **int** x1, **int** x2, **char** *newName, FILE *pFile)

This function checks if new matrix already exist in file array. If it is it's prints error. And check's sliced matrix start stop row and columns. If everything okays it slice matrix and creates new matrix.

void matslicecol(**struct** file *filearray, **int** *file_count, **char** *name, **int** index, **int** start, **int** stop, **char** *newName,FILE *pFile)

This function checks if new vector already exist in file array. If it is it's prints error. And its slice a vector in matrix column.

void matslicerow(**struct** file *filearray, **int** *file_count, **char** *name, **int** index, **int** start, **int** stop, **char** *newName, FILE *pFile)

This function checks if new vector already exist in file array. If it is it's prints error. And its slice a vector in matrix row..

void add(**struct** file *filearray, **int** *file_count, **char** *matrix1, **char** *matrix2, FILE *pFile)

This function checks matrix1 and matrix2 row and column if they are not equal it prints error. Otherwise its add matrix2 to matrix1 and update matrix1.

void multiply(**struct** file *filearray, **int** *file_count, **char** *matrix1, **char** *matrix2, **int** **copy, FILE *pFile)

This function checks matrix1 and matrix2 row and column if they are not equal it prints error. Otherwise its multiply matrix1 and matrix2 then update matrix1.

void subtract(**struct** file *filearray, **int** *file_count, **char** *matrix1, **char** *matrix2, FILE *pFile)

This function checks matrix1 and matrix2 row and column if they are not equal it prints error. Otherwise its subtract matrix1 to matrix2 and update matrix1.

void readCommand(**char** *v[], **struct** file *filearray, **int** *file_count, **int** **copy, FILE *pFile)

This function reads command file. İt checks mistakes and send parameters to function.

MY ALGORİTM

My program starts with creating 2d array named of Copy. I Use copy array because while I trying to realloc to file Struct 2d array It adress goes to another adres in memory and I lost some element values. After I create filearray array for create structs dynamically. Then ı open my output file. After that I call readCommand. This function read commands and calls function which ı need to perform operations for it. When command file operations finished. I free my struct elements, struct array, and copy array and file pointers.