T.C. MARMARA ÜNİVERSİTESİ FİZİK BÖLÜMÜ

İLERİ PROGRAMLAMA

ÖDEV RAPORU

KONU: MatMul

Github : MatMul

Begimai Saadakova (120514906)

TESLİM TARİHİ: 2016/12/26

**Purpose:** It needs to create two folders as **MatrisMulTest.c** and **MatrisMul.c**.At folder  **MatrisMul.c** there are the functions read the given matrices,multiply these matrices,print them on screen and empty them.

**Summary:** I needed to create matrix with N size of rows and columns,for that i took function malloc in order to allocate the memory from operating memory with size integer. Made cycle of rows and in each row there is a cycle of columns.Into this matrix need to enter the datas and then it can be printed out by fuction **print\_matrix** which also has two iteration inside of each other.For multiplication need another matrix with the size N, and I used calloc function to get enough space for the results of two multiplied matrices. Then need to itterate the array and initialize it’s elements to zero. At multiplication of matrix it is important to be equal the number of column of first matrix to row of the second, variable k does this work. Now we can multiply the matrices and store the result in new matrix. Also need to free created matrices after every usage.

**#include <stdio.h>**

**#include "MatrisMul.h"**

int \*\*MatrisMatMul(int \*\*matrix1, int \*\*matrix2,int row1,int column1,int column2) ;

int \*\*read\_matrix(int rows, int cols);

void print\_matrix(int rows, int cols,int \*\*mat);

void free\_matrix(int \*\*mat);

int main() {

int i, j;

int \*\*A,\*\*B,\*\*res; /\*name of the matrices\*/

int row1, col1, row2, col2; /\*rows and columns of the fisrt and second matrices\*/

printf("Enter # of rows of matrix1:");

scanf("%d",&row1);

printf("Enter # of cols of matrix1:");

scanf("%d",&col1);

printf("Enter # of rows of matrix2:");

scanf("%d",&row2);

printf("Enter # of cols of matrix1:");

scanf("%d",&col2);

if(col1!=row2) { /\*If column of first matrix in not equal to row of second matrix, asking user to

enter the size of matrix again.\*/

printf("Invalid Order of matrix.\n");

printf("Enter # of rows of matrix1:");

scanf("%d",&row1);

printf("Enter # of cols of matrix1:");

scanf("%d",&col1);

printf("Enter # of rows of matrix2:");

scanf("%d",&row2);

printf("Enter # of cols of matrix1:");

scanf("%d",&col2);

}

printf("Matrix, enter %d elements for matrix1: \n",row1\*col1); /\*Input data for the matrices\*/

A = read\_matrix(row1,col1);

printf("Matrix, enter %d elements for matrsix2: \n",row2\*col2);

B = read\_matrix(row2,col2);

printf("Your Matrix A:\n"); /\* Print the entered data \*/

print\_matrix(row1,col1,A);

printf("Your Matrix B:\n");

print\_matrix(row2,col2,B);

res=MatrisMatMul(A,B,row1,col1,col2) ; /\*result of the multiplication\*/

printf("\nThe Multiplication of two matrix is:\n");

print\_matrix(row1,col2,res);

free\_matrix(A); /\* Free the matrices \*/

free\_matrix(B);

free\_matrix(res);

return 0;

}

**#ifndef MatrisMul\_H**

**#define MatrisMul\_H**

int \*\*MatrisMatMul(int \*\*matrix1, int \*\*matrix2,int row1,int column1,int column2); /\*prototype for multiplication of

two matrix\*/

int \*\*read\_matrix(int rows, int cols); /\*prototype to read the given matrix\*/

void print\_matrix(int rows, int cols,int \*\*mat); /\*prototype to print out the matrix\*/

void free\_matrix(int \*\*mat); /\*prototype to empty value of the matrix\*/

**#endif**

**#include <stdio.h>**

**#include <stdlib.h>**

int **\*\*MatrisMatMul** (int \*\*matrix1, int \*\*matrix2,int row1,int column1,int column2) {

int i,j,k;

int \*\*result=(int\*\*)malloc(sizeof(int\*),row1); /\*allocate memory for arrays with size of integers\*/

for(i=0;i<column2;i++)

result[i]=(int\*)malloc(sizeof(int,column2));

for(i=0;i<row1;i++) {

for(j=0;j<column2;j++) {

result[i][j]=0; /\*Initializing elements of matrix to 0\*/

for(k=0;k<column1;k++)

result[i][j]+= matrix1[i][k]\*matrix2[k][j]; /\*Multiplying matrix1 and matrix2 and storing in array result.\*/

}

printf("\n");

}

return result;

}

int **\*\*read\_matrix** (int rows, int cols) {

int i,j;

int \*\*mat = (int \*\*) malloc(sizeof(int\*)\*rows); /\* allocate matrix\*/

for(i=0; i<rows; i++)

mat[i] = (int\*) malloc(sizeof(int)\*cols);

for(i=0; i<rows; i++) {

for(j=0; j<cols; j++) {

printf("Enter matrix[%d][%d]=",i,j);

scanf("%d",&mat[i][j]);

}

}

return mat;

}

void **print\_matrix** (int rows, int cols,int \*\*mat) {

int i=0,j=0;

for(i=0; i<rows; i++) { /\* Iterate of each row \*/

for(j=0; j<cols; j++) { /\* In each row, go over each col element \*/

printf("%d\t",mat[i][j]); /\* Print each row element \*/

}

printf("\n");

}

}

void **free\_matrix** (int \*\*mat) { /\*empty the matrix\*/

int i,rows;

for(i=0;i<rows;i++)

free(mat[i]);

free(mat);

}