Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

# Кафедра радіотехнічних систем

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4**

з дисципліни: «Інформатика1. Основи програмування та алгоритми»

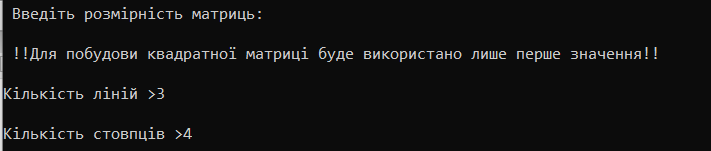
|  |  |
| --- | --- |
|  | Виконав: Черкас Максим Віталійович  Група: РС-21  Викладач: Турєєва О.В. |

Київ – 2023

**Мета роботи**: вдосконалення знань, умінь та навичок з технології розроблення програмного забезпечення (ПЗ) з використанням мови С у процедурній парадигмі. Також лабораторна робота дає основні навички обчислення нелінійних рівнянь декількома способами.

**Ключові моменти**:

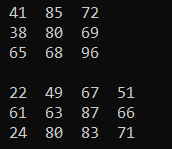
1. На початку програми підключаємо бібліотеки та створюємо прототипи.
2. Далі ініціалізуємо змінні та вводимо розмірність матриці.



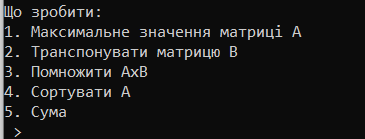
1. Даємо користувачеві вибір як заповнити матрицю(автоматично чи самостійно).



1. Виводимо матрицю користувачу на екран.



1. Вибираємо подальші дії з 5 варіантів.



1. Обравши варіант певний варіант вказівник напрямляє комп’ютер до алгоритму який виконує певні дії.
2. В кожному варіанті передбачено вивід результату на екран.

**Код:**

#include <stdio.h>

#include <stdlib.h>

void print\_received\_arrays( int \*\*A, int \*\*B, int M, int N);

int max(int \*\*A,int M,int N);

int transpon (int \*\*B,int \*\*T,int M,int N);

int product(int \*\*A,int \*\*B,int \*\*Prod,int M,int N);

int sort(int \*\*A,int M,int N);

int matrix\_sum(int \*\*A,int \*\*B,int sum,int M,int N);

int main()

{

int input\_var;

int action;

int sum;

int M=3;

int N=4;

printf("\n Введіть розмірність матриць:");

printf("\n\n !!Для побудови квадратної матриці буде використано лише перше значення!!\n");

printf("\nКількість ліній >");

scanf("%d", &M);

printf("\nКількість стовпців >");

scanf("%d", &N);

int \*\*A = NULL;

A=(int\*\*) malloc(M \* sizeof(int));

for(int i=0;i<M;i++){

A[i]=(int\*) malloc(M \* sizeof(int));

}

int \*\*B=NULL;

B=(int\*\*) malloc(N \* sizeof(int));

int \*\*Prod=NULL;

Prod=(int\*\*) malloc(N \* sizeof(int));

int \*\*T=NULL;

T=(int\*\*) malloc(M \* sizeof(int));

for(int i=0;i<N;i++){

B[i]=(int\*) malloc(M \* sizeof(int));

Prod[i]=(int\*) malloc(M \* sizeof(int));

T[i]=(int\*) malloc(M \* sizeof(int));

}

do{

system("cls");

printf(" Заповніть матрицю: 1. Автоматично 2. Самому\n>");

scanf("%d",&input\_var);

}while(input\_var != 1 && input\_var != 2);

if(input\_var==2){

printf("Квадратна матриця A:\n");

for (int i=0;i<M;i++)

{

for (int j=0;j<M;j++){

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

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

}

}

printf("Прямокутна матриця B:\n");

for (int i=0;i<M;i++)

{

for (int j=0;j<N;j++){

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

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

}

}

}

else{

for (int i=0;i<M;i++){

for (int j=0;j<M;j++){

A[i][j]= rand()%101;

}

}

for (int i=0;i<M;i++){

for (int j=0;j<N;j++){

B[i][j]= rand()%101;

}

}

}

print\_received\_arrays(A, B, M, N);

do{

printf("\nЩо зробити:\n1. Максимальне значення матриці A \n2. Транспонувати матрицю B\n3. Помножити AxB\n4. Сортувати A\n5. Сума\n >");

scanf("%d",&action);

}while(action != 1 && action != 2 && action != 3 && action != 4 && action != 5);

switch (action){

case 1:

max(A,M,N);

break;

case 2:

transpon(B,T,M,N);

break;

case 3:

printf("AxB = \n");

product(A,B,Prod,M,N);

break;

case 4:

sort(A,M,N);

break;

case 5:

matrix\_sum(A,B,sum,M,N);

break;

}

for(int i=0;i<1;i++){

free(A[i]);

}

free(A);

for(int i=0;i<N;i++){

free(B[i]);

free(Prod[i]);

free(T[i]);

}

free(B);

free(Prod);

free(T);

return 0;

}

//------------------------------------------------------------------------------------------

void print\_received\_arrays( int \*\*A, int \*\*B, int M, int N){

for (int i=0;i<M;i++){

for (int j=0;j<M;j++)

{

printf("%3d ", A[i][j]);

}

printf("\n");

}

printf("\n");

for (int i=0;i<M;i++){

for (int j=0;j<N;j++){

printf("%3d ", B[i][j]);

}

printf("\n");

}

}

//------------------------------------------------------------------------------------------

int max(int \*\*A,int M,int N){

int maximal= 0;

for(int i=0;i<M;i++){

for(int j=0;j<M;j++){

if (A[i][j] > maximal){

maximal = A[i][j];

}

}

}

printf("\nМаксимальне значення A: %d\n",maximal);

}

//-------------------------------------------------------------------------------------------

int transpon (int \*\*B,int \*\*T,int M,int N){

for (int i=0;i<M;i++){

for (int j=0;j<N;j++){

T[j][i] = B[i][j];

}

}

printf("\n Транспонована матриця B : \n");

for (int i=0;i<N;i++){

for (int j=0;j<M;j++){

printf("%3d ", T[i][j]);

}

printf("\n");

}

}

//-----------------------------------------------------------------------------------------

int product(int \*\*A,int \*\*B,int \*\*Prod,int M,int N){

for(int i=0;i<M;i++){

for(int j=0;j<N;j++){

Prod[i][j]=0;

for(int p=0;p<M;p++){

Prod[i][j] += A[i][p] \* B[p][j];

}

}

}

for (int i=0;i<M;i++){

for (int j=0;j<N;j++)

{

printf("%3d ", Prod[i][j]);

}

printf("\n");

}

}

//---------------------------------------------------------------------------------------

int sort(int \*\*A,int M,int N){

int sort\_var;

do{

printf("\n Сортувати: 1. матрицю 2. лінію матриці\n >");

scanf("%d", &sort\_var);

}while(sort\_var !=1 && sort\_var !=2);

if (sort\_var == 1){

int temp;

for(int k = 0; k < M \* N; ++k) {

for(int i = 0; i < M; ++i) {

for(int j = 0; j < N; ++j) {

if(j != N - 1) {

if(A[i][j+1] < A[i][j]) {

int temp = A[i][j+1];

A[i][j+1] = A[i][j];

A[i][j] = temp;

}

}

}

}

}

for (int i=0;i<M;i++){

for (int j=0;j<M;j++){

printf("%3d ", A[i][j]);

}

printf("\n");

}

}

if(sort\_var == 2){

int sort = 0;

int temp;

printf("\nВведіть рядок для сортування: ");

scanf("%d",&sort);

for(int k=1;k<M;k++)

{

for(int j=0;j<M-k;j++){

if(A[sort][j]>A[sort][j+1]){

temp = A[sort][j];

A[sort][j] = A[sort][j+1];

A[sort][j+1] = temp;

}

}

}

for(int j=0;j<M;j++){

printf("%3d ",A[sort][j]);

}

}

}

//----------------------------------------------------------------------------------------

int matrix\_sum(int \*\*A,int \*\*B,int sum,int M,int N){

for(int i=0;i<M;i++){

int sum=0;

for(int j=0;j<N;j++){

sum = sum + A[i][j];

}

printf("\n Сума ліній:%d A =%d",i,sum);

}

for(int j=0;j<N;j++){

int sum=0;

for(int i=0;i<M;i++){

sum = sum + B[i][j];

}

printf("\n Сума стовпців:%d B =%d",j,sum);

}

}

***Висновки:***

На цій лабораторній роботі ми навчилися працювати з вказівниками та матрицями. Удосконалили вміння розроблювати алгоритми.

