Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4**

з дисципліни: «Інформатика 1»

|  |  |
| --- | --- |
|  | Виконав : Рокунець Андрій Олександрович  Група: РЕ-12  Викладачі: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

Код:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <conio.h>

#include <time.h>

void printArray(double \*\*ptr, int n, int m);

double maximum(double \*\*ptr, int n, int m);

double minimum(double \*\*ptr, int n, int m);

double \*\* dobutok(double \*\*ptrA, int nA, int mA, double \*\*ptrB, int nB, int mB);

void TRANS\_MATRIC(double \*\*ptr, int n, int m, double \*\*ptrT);

void SORT\_ROW(double \*ptr, int n);

void SUMA\_ROW(double\*\*ptr,int n,int m);

void SUMA\_COL(double\*\*ptr,int n,int m);

int main()

{

int nA, nB, mB;

int localVar, Variant;

double \*\*ptrA, \*\*ptrB, \*\*ptrT, \*\*ptrAB;

printf(" nA= ");

scanf("%u",&nA);

printf(" nB= ");

scanf("%u",&nB);

printf(" mB= ");

scanf("%u",&mB);

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

ptrA=malloc(nA\*sizeof(double\*));

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

{

ptrA[i]=malloc(nA\*sizeof(double));

}

printf(" A:1-auto, 2-enter : ");

scanf("%u",&localVar);

if(localVar==1)

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

{

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

ptrA[i][j] = -10 + rand() % 21;

}

else

{

printf("enter matric elements A\n");

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

{

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

{

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

scanf("%lf",&ptrA[i][j]);

}

}

}

printArray(ptrA,nA,nA);

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

ptrB=malloc(nB\*sizeof(double\*));

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

{

ptrB[i]=malloc(mB\*sizeof(double));

}

printf(" B :1-auto, 2-enter : ");

scanf("%u",&localVar);

if(localVar==1)

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

{

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

ptrB[i][j]= -20 + rand() % 55;

}

else

{

printf(" enter matric elements B\n");

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

{

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

{

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

scanf("%lf",&ptrB[i][j]);

}

}

}

printArray(ptrB,nB,mB);

printf("1-max and min element\n\n 2-Trans matrica\n\n 3-dobutok matrix A and B\n\n 4-sortuvania matricy A\n\n 5- sum ROW A,sum COL B\n\n");

printf("Variant=");

scanf("%u", &Variant);

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

if(Variant==1){

double maxA= maximum(ptrA,nA,nA);

printf(" max number A = %.2lf\n",maxA);

double minA=minimum(ptrA,nA,nA);

printf(" min number A = %.2lf\n",minA);

}

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

if(Variant==2){

ptrT=malloc(mB\*sizeof(double\*));

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

{

ptrT[i]=malloc(nB\*sizeof(double));

}

TRANS\_MATRIC(ptrB,nB,mB, ptrT);

printf("transponovana matryca B\n");

printArray(ptrT,mB,nB);}

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

if(Variant==3){

ptrAB=dobutok(ptrA,nA,nA,ptrB,nB,mB);

printf("dobutok matric A\*B\n");

if(ptrAB!=NULL)

printArray(ptrAB,nA,mB);}

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

if(Variant==4){

int row=0;

printf("Enter row for sort ");

scanf("%u",&row);

SORT\_ROW(ptrA[row],nA);

printArray(ptrA,nA,nA);}

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

if(Variant==5){

printf("suma ROWS A\n");

SUMA\_ROW(ptrA,nA,nA);

printf("suma COLS B\n");

SUMA\_COL(ptrB,nB,mB);

}

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

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

{

free(ptrA[i]);

}

free(ptrA);

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

{

free(ptrB[i]);

}

free(ptrB);

if(ptrT!=0)

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

{

free(ptrT[i]);

}

free(ptrT);

}

if(ptrAB!=0)

{

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

{

free(ptrAB[i]);

}

free(ptrAB);

}

return 0;

}

b

void printArray(double\*\*ptr,int n, int m){

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

{

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

printf("%7.2lf",ptr[i][j]);

printf("\n");

}

}

double maximum(double\*\*ptr,int n, int m){

double max=ptr[0][0];

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

{

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

if(ptr[i][j]>max){

max=ptr[i][j];

}

}

return max;

}

double minimum(double\*\*ptr,int n, int m){

double min=ptr[0][0];

int var;

printf("Choose: 1 -over diagonal; 2- under diagonal");

scanf("%u",&var);

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

{

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

{

if(var==1&& i<j && ptr[i][j]<min)

min=ptr[i][j];

if(var==2&& i>j && ptr[i][j]<min)

min=ptr[i][j];

}

}

printf("min element ");

if(var==1)

printf("over diagonal ");

if(var==2)

printf("under diagonal ");

printf("%.2lf \n", min);

return min;

}

double \*\* dobutok(double\*\*ptrA,int nA, int mA, double\*\*ptrB,int nB, int mB){

if(mA!=nB)

{

printf("Error ");

return NULL;

}

else{

double \*\*ptrAB;

ptrAB=malloc(nA\*sizeof(double\*));

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

{

ptrAB[i]=malloc(mB\*sizeof(double));

}

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

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

ptrAB[i][j]=0;

for(int k=0; k<mA; k++)

ptrAB[i][j]+=ptrA[i][k]\*ptrB[k][j];

}

return ptrAB;

}

}

void TRANS\_MATRIC(double\*\*ptr,int n,int m, double\*\* ptrT)

{

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

{

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

{

ptrT[j][i]=ptr[i][j];

}

}

}

void SORT\_ROW(double\*ptr,int n)

{

double temp;

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

{

for(int j=0; j<n-i-1; j++)

{

if(ptr[j] > ptr[j+1])

{

temp=ptr[j];

ptr[j]=ptr[j+1];

ptr[j+1]=temp;

}

}

}

}

void SUMA\_ROW(double\*\*ptr,int n,int m)

{

double sum;

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

{

sum=0;

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

{

sum+=ptr[i][j];

}

printf("suma %u row=%.2lf\n",i,sum);

}

}

void SUMA\_COL(double\*\*ptr,int n,int m)

{

double sum;

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

{

sum=0;

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

{

sum+=ptr[i][j];

}

printf("suma %u col=%.2lf\n",j,sum);

}

}

