Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4**

з дисципліни: «Інформатика. Основи програмування та алгоритми»

Виконав: Кривунець Максим Віталійович

Група: РЕ–11

Викладач: доц. Катін П. Ю.

Київ – 2021

Мета роботи: навчитися працювати з одновимірними та двовимірними масивами,адресами та вказівниками.

Код:#include <stdio.h>

#include <stdlib.h>

void printArr(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 maxelemet(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 minelemet(double\*\*ptr,int n, int m){

double min=ptr[0][0];

int menu;

printf("vuberit: 1 -vyche diagonali; 2- nyche diagonali");

scanf("%u",&menu);

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

{

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

{

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

min=ptr[i][j];

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

min=ptr[i][j];

}

}

printf("minimalnu element ");

if(menu==1) printf("vyche diagonali ");

if(menu==2) printf("nyche diagonali ");

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

return min;

}

double \*\* multmatrix(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 transpon(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 sortrow(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 sumarow(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 sumacol(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);

}

}

int main()

{

int nA, nB, mB;

int menu, MENU=1;

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

printf("vedit nA ");

scanf("%u",&nA);

printf("vedit nB ");

scanf("%u",&nB);

printf("vedit mB ");

scanf("%u",&mB);

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

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

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

{

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

}

printf("inicializacia matryc A:1-formula, 2-vruchnu : ");

scanf("%u",&menu);

if(menu==1)

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

{

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

ptrA[i][j]=i+2\*j;

}

else

{

printf("vedit elementu matryci 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]);

}

}

}

printf("vuvid matryci A\n");

printArr(ptrA,nA,nA);

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

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

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

{

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

}

printf("inicializacia matrycy B :1-formula, 2-vruchnu : ");

scanf("%u",&menu);

if(menu==1)

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

{

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

ptrB[i][j]=2\*i+3\*j;

}

else

{

printf("vedit elementu matryci 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]);

}

}

}

printf("vuvid matryci B\n");

printArr(ptrB,nB,mB);

while(MENU!=0)

{

printf("--------------MENU------------------\n");

printf("1-poshuk max/min element\n 2-transponuvania matryci\n 3-mnozenia matryci A\*B\n ");

printf("4-sortuvania matryci A\n 5-suma ryadkiv A, suma colonok B\n 0-exit\n");

printf("? ");

scanf("%u",&MENU);

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

//zavd1

if(MENU==1){

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

printf("maxelement matryci A = %.2lf\n",maxA);

double minA=minelemet(ptrA,nA,nA);}

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

//zavd2

if(MENU==2){

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

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

{

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

}

transpon(ptrB,nB,mB, ptrT);

printf("transponovana matryca B\n");

printArr(ptrT,mB,nB);}

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

//zavd 3

if(MENU==3){

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

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

if(ptrAB!=NULL)

printArr(ptrAB,nA,mB);}

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

//zavd 4

if(MENU==4){

int row=0;

printf("vedit nomer ryadka dlya sortuvania ");

scanf("%u",&row);

sortrow(ptrA[row],nA);

printf("sorted matryca\n");

printArr(ptrA,nA,nA);}

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

//zavd 5

if(MENU==5){

printf("suma ryadkiv matryci A\n");

sumarow(ptrA,nA,nA);

printf("suma colonki matryci B\n");

sumacol(ptrB,nB,mB);

}

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

} // end while

//звільненя памяті

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;

}

Висновок: Ми навчилися працювати з динамічними масивами та обчислювати матриці за допомогою масивів.