Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ**

**№4**

з дисципліни:

«ІНФОРМАТИКА 1»

|  |  |
| --- | --- |
|  | Виконав: Грабун Владислав  Група: РЕ–12  Викладачі: Катін П.Ю.  (посада, вчене звання, науковий ступінь, прізвище та ініціали)  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

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

#include <stdio.h>

#include <stdlib.h>

int i,j,temp,N=3,M=4,choise;

int maximal=0,sort,sum=0,vvod,end;

int Max (int \*\*A);

int Transponate (int \*\*B,int \*\*T);

int Multiply(int \*\*A,int \*\*B,int \*\*Dob);

int Sort(int \*\*A,int sort);

int Summ(int \*\*A,int \*\*B,int sum);

int main()

{

printf("Select a method to create a matrix:\n1. Random\n2. From keyboard\n");

scanf("%d",&vvod);

if (vvod==2)

{

printf("\nEnter the size of the square matrix: ");

scanf("%d",&N);

printf("\nEnter the size of the rectangular matrix: ");

scanf("%d",&M);

}

int \*\*A=NULL;

A=(int\*\*)calloc(N,sizeof(int));

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

{

A[i]=(int\*)calloc(N,sizeof(int));

}

int \*\*B=NULL;

B=(int\*\*)calloc(M,sizeof(int));

int \*\*Dob=NULL;

Dob=(int\*\*)calloc(M,sizeof(int));

int \*\*T=NULL;

T=(int\*\*)calloc(N,sizeof(int));

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

{

B[i]=(int\*)calloc(N,sizeof(int));

Dob[i]=(int\*)calloc(N,sizeof(int));

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

}

if(vvod==2)

{

printf("Enter the elements of a square matrix:\n");

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

{

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

{

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

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

}

}

printf("Introduce elements of rectangular matrix:\n");

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

{

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

{

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

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

}

}

}

else

{

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

{

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

{

A[i][j]=rand() %20;

}

}

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

{

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

{

B[i][j]=rand() %15;

}

}

}

printf("\nThe initial matrix A and B:\n");

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

{

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

{

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

}

printf("\n");

}

printf("\n");

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

{

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

{

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

}

printf("\n");

}

printf("\nUsage menu:\n1. The maximum value of A \n2. Transpose B\n3. Multiply AxB\n4. Sort matrix row A\n5. Show the amount in rows and columns\n");

scanf("%d",&choise);

switch (choise)

{

case 1:

Max (A);

break;

case 2:

Transponate(B,T);

break;

case 3:

printf("AxB:\n");

Multiply(A,B,Dob);

break;

case 4:

printf("\nEnter the line number to sort: ");

scanf("%d",&sort);

Sort(A,sort);

break;

case 5:

Summ(A,B,sum);

break;

}

printf("\nWhat are the next steps?\n1. Restart\n2. Exit\n");

scanf("%d",&end);

if(end==1){

system("cls");

return main(); }

else

return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int Max (int \*\*A)

{

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

{

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

{

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

maximal=A[i][j];

}

}

printf("The maximum value of the matrix А: %d\n",maximal);

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int Transponate (int \*\*B,int \*\*T)

{

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

{

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

{

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

}

}

printf("\n Transposed matrix : is:\n");

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

{

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

{

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

}

printf("\n");

}

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int Multiply(int \*\*A,int \*\*B,int \*\*Dob)

{

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

{

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

{

Dob[i][j]=0;

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

{

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

}

}

}

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

{

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

{

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

}

printf("\n");

}

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int Sort(int \*\*A,int sort)

{

for(i=0;i<N-1;i++)

{

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

{

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

}

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int Summ(int \*\*A,int \*\*B,int sum)

{

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

{

sum=0;

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

{

sum+=A[i][j];

}

printf("\n%d The sum of the elements of the rows of the matrix А =%d",i,sum);

}

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

{

sum=0;

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

{

sum+=B[i][j];

}

printf("\n%d The sum of the elements of the matrix columns В =%d",j,sum);

}

}

Висновок:

Матриці А (прямокутна), та B (квадратна)

Ми написали програми, що виконують певні дії с массивами – множення, транспонування, сортування, знаходження найбільшого числа у певній строчці