****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Calancea Cătălin**

**MI-222**

**Raport**

**pentru lucrarea de laborator Nr.5**

***la cursul de “Programarea calculatoarelor”***

Verificat:

**Guțu Maria,** *doctor, conf. univ.*

Departamentul Informatică şi IS,

Facultatea FCIM, UTM

**Chișinău – 2022**

**Scopul lucrării:** Studierea posibilităţilor şi mijloacelor limbajului C pentru programarea algoritmilor cu structură ramificată şi ciclică la prelucrarea tablourilor bidimensionale.

**Nr. variantei:** 3

**Condiția problemei ( sarcinii de lucru ) :**

1. Să se determine produsul elementelor mai mici decât media aritmetică a tuturor elementelor, precum pozițiile lor.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m;

float avg=0.0 , result=1.0;

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

sum+=A[i][j];

}}

avg=(float)sum/(n\*m);

printf("Average = %.2lf ",avg);

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

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

if(A[i][j] < avg){

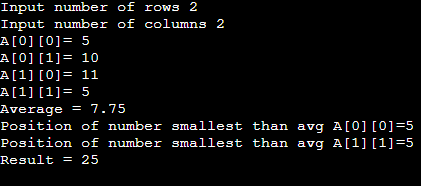
result\*=A[i][j];

printf("\nPosition of number smallest than avg A[%d][%d]=%d ",i,j,A[i][j]);}}}

printf("\nResult = %2.lf ", result);

return 0;

}



**Condiția problemei ( sarcinii de lucru ) :**

1. Să se determine valoarea minimală pozitivă dintre elementele tabloului şi poziţia primului element cu această valoare, precum şi numărul elementelor pozitive în tablou.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m,reference=0,count=0 ;

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

if(A[i][j]>0){

count++;}

if (A[i][j]<0){

A[i][j]=32767;

}

}}

int min=A[0][0],i,j;

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

{

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

{

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

min= A[i][j];

}

}

printf("\nSmallest number is :%d" ,min);

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

{

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

{

if(A[i][j]==min)

printf("'\nFirst position for smallest positive number is A[%d][%d]" ,i,j);

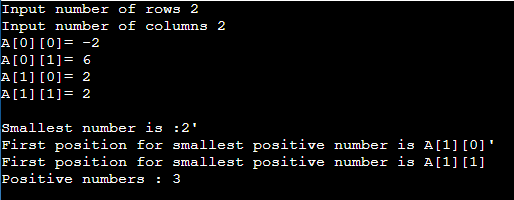
}

}

printf ("\nPositive numbers : %d ",count);

return 0;

}



**Condiția problemei ( sarcinii de lucru ) :**

1. Se consideră doi vectori cu câte n și, respectiv, m componente numere întregi. Să se creeze un al treilea vector care va conține toate elementele impare a primului și celui de al doilea vector aranjate în ordine crescătoare/descrescătoare.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m,B[50],k=0 ,product=1;

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

}}

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

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

if (A[i][j]<0){

B[k]=A[i][j];

k++;

product\*=B[i];

printf("\nB[%d]= %d ",k-1,B[i]);

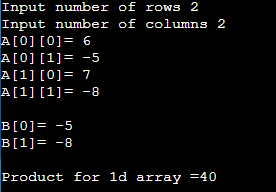
}}}

printf("\n");

printf("\nProduct for 1d array =%d",product);

return 0;

}



**Condiția problemei ( sarcinii de lucru ) :**

1. Să se zerografieze diagonala principală și să se determine suma elementelor ce se află sub diagonala principală.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m;

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

}}

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

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

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

}

printf("\n");

}

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

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

if (i==j){

A[i][j]=0;

}

}}

printf("\n");

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

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

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

}

printf("\n");

}

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

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

if (i>j){

sum +=A[i][j];

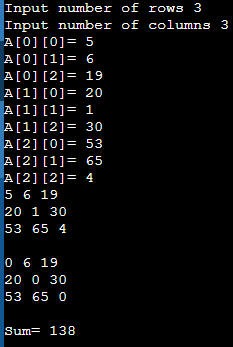
}

}}

printf ("\nSum= %d ",sum);

return 0;

}



**Condiția problemei ( sarcinii de lucru ) :**

1. Să se determine suma elementelor ce se află deasupra diagonalei secundare, precum și media aritmetică a elementelor ce se află deasupra diagonalei secundare și, totodată, sub diagonala principală.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m;

float avg=0.0;

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

}}

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

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

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

}

printf("\n");

}

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

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

if ((i+j)<(n-1)){

sum+=A[i][j];

}

}}

printf("\nSum=%d",sum);

printf("\n");

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

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

if ((i+j)<(n-1) && (i>j)){

avg=(float)sum/(n\*m);

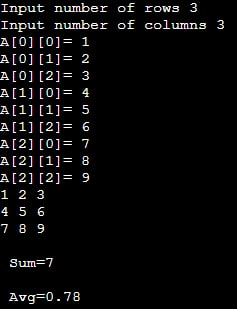
}

}}

printf("\nAvg=%.2lf",avg);

return 0;

}

****

**Condiția problemei ( sarcinii de lucru ) :**

1. Să se determine valoarea minimală pe fiecare linie a elementelor tabloului şi poziţiile elementelor cu această valoare, precum şi produsul tuturor elementelor negative din tablou.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m,product=1,min=A[0][0];

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

if(A[i][j]<0){

product\*=A[i][j];}

}}

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

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

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

}

printf("\n");

}

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

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

min=A[i][j];

for(int k=0; k<n; k++){

if (A[i][k] < min ){

min=A[i][k];

}

if (A[i][j]==min){

printf("\nPosition of smallest element A[%d][%d]" ,i,j);

}

}

}

printf("\nSmallest element on %d row is : %d ", i+1, min);

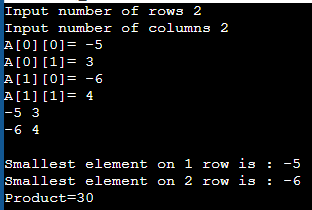
}

printf("\n");

printf("Product=%d",product);

return 0;

}

****

**Condiția problemei ( sarcinii de lucru ) :**

1. Să se determine valoarea maximală pe fiecare linie a elementelor tabloului și această valoare să se interschimbe cu elementul de pe diagonala secundară a aceleiași linii. Prin urmare, în final, diagonala secundară va conține valoarea maximală a fiecărei linii.

**Codul programului:**

#include <stdio.h>

int main()

{

int A[50][50],sum=0,n,m,product=1,max=A[0][0],aux=0;

printf("Input number of rows ");

scanf("%d",&n);

printf("Input number of columns ");

scanf("%d",&m);

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

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

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

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

if(A[i][j]<0){

product\*=A[i][j];}

}}

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

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

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

}

printf("\n");

}

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

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

max=A[i][j];

for(int k=0; k<n; k++){

if (A[i][k] > max ){

max=A[i][k];

}

}

}

printf("\nBigger element on %d row is : %d ", i+1, max);

}

printf("\n");

return 0;

}