II variant. 2olculu masssivler

1) NxN ölçülü iki A və B matrisi verilmişdir. Onlardan birincini ilk N sütuna, ikincini isə növbəti sütunlara yerləşdirməklə Nx2N ölçülü massiv tərtib etməli.

#include <iostream>

using namespace std;

int main()

{

int A[2][2]={

{1,2},

{3,4}};

int B[2][2]={

{10,20},

{30,40}};

int resultArray[2][4]={};

for(int i=0;i<2;i++){

for(int j=0;j<2;j++){

resultArray[i][j]=A[i][j];

}

for(int j=2;j<4;j++){

resultArray[i][j]=B[i][j-2];

}

}

for(int i=0;i<2;i++){

for(int j=0;j<4;j++){

cout<<resultArray[i][j]<<" ";

}

cout<<endl;

}

return 0;

}

2) A(n,n) massivində üçüncü və dördüncü sətirlərin yerini dəyişməli

#include <iostream>

using namespace std;

int main()

{

int A[4][4]={

{1,2,3,4},

{3,4,5,6},

{10,20,30,40},

{30,40,50,60}

};

int resultArray[4][4]={};

for(int i=0;i<2;i++){

for(int j=0;j<4;j++){

resultArray[i][j]=A[i][j];

}

}

for(int i=0;i<4;i++){

resultArray[2][i]=A[3][i];

resultArray[3][i]=A[2][i];

}

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

cout<<resultArray[i][j]<<" ";

}

cout<<endl;

}

return 0;

}

3) X(n,m) massivində iki ilk mənfi elementlər arasındakı məsafəni tapmalı

#include <iostream>

using namespace std;

int main()

{

int A[3][4]={

{1,-2,3,4},

{3,-4,5,6},

{30,40,50,-60}

};

int first=0;

int second=0;

for(int i=0;i<3;i++){

for(int j=0;j<4;j++){

if(first==0 && A[i][j]<0){

first=10\*i+j;

}

else if(second==0 && A[i][j]<0 && first!=0){

second=10\*i+j;

}

}

}

int counter=0;

for(int i=first/10;i<(second/10);i++){

for(int j=first%10+1;j<4;j++){

counter++;

}

for(int j=0;j<second%10;j++){

counter++;

}

}

cout<<counter;

return 0;

}

4) X(n,n) massivində baş diaqonala qonşu diaqonalları çapa çıxarmalı

#include <iostream>

using namespace std;

int main()

{

int A[3][3]={

{1,2,4},

{3,4,5},

{7,8,9}

};

int newArray[3];

int index=0;

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

if(i==2-j){

newArray[index++]=A[i][j];

}

}

}

index=0;

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

if(i==j){

A[i][j]=newArray[index++];

}

}

}

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

cout<<A[i][j]<<" ";

}

cout<<endl;

}

return 0;

}

5) X(n,m) massivinin sütunlarını artma sırasına görə nizamlamalı

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int size) {

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

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

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main()

{

int A[3][3]={

{1,-2,4},

{3,10,5},

{9,8,3}

};

int newArray[3];

int index=0;

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

bubbleSort(A[i],3);

}

}

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

cout<<A[i][j]<<" ";

}

cout<<endl;

}

return 0;

}

6) X(n,m) və Y(n,m) massivlərində mövcud olan eyni sətirləri çapa çıxarmalı

#include <iostream>

using namespace std;

int main()

{

int A[3][4]={

{1,2,3,4},

{3,4,69,43}};

int B[3][4]={

{30,40,80,67},

{1,2,3,4}};

for(int k=0;k<3;k++){

for(int i=0;i<3;i++){

for(int j=0;j<4;j++){

if(A[i][j]==B[k][j]){

cout<<A[i][j]<<" ";

}

else{

break;

}

}

}

}

return 0;

}

7) X(n,m) massivində (n,m) elementlərindən başlayaraq bütün elementləri sütunlar üzrə pilləvari çapa çıxarmalı

8) X(n,m) massivində özündən əvvəlki elementlərin cəminə bərabər olan elementin yerləşdiyi sətiri çapa çıxarmalı

#include <iostream>

using namespace std;

int main()

{

int A[2][4]={

{1,2,3,6},

{3,4,69,43}};

int row=-1;

for(int i=0;i<2;i++){

int sum=0;

for(int j=0;j<3;j++){

sum+=A[i][j];

}

if(sum==A[i][3]){

row=i;

}

}

for(int i=0;i<4;i++){

cout<<A[row][i]<<" ";

}

return 0;

}

9) X(n,m) massivində yalnız mənfi elementlərdən ibarət sətirdə minimum elementi tapmalı

#include <iostream>

using namespace std;

int main()

{

int A[2][4]={

{1,-100,3,6},

{-3,-4,-69,-43}};

int row=-1;

for(int i=0;i<2;i++){

for(int j=0;j<4;j++){

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

row=i;

}

else{

row=-1;

break;

}

}

}

int minElement=A[row][0];

for(inti=0;i<4;i++){

for(int j=0;j<4;j++){

if(A[row][j]<minElement){

minElement=A[row][j];

}

}

}

cout<<minElement;

return 0;

}

10) X(n,m) massivində ilk və son mənfi elementlər arasındakı məsafəni tapmalı

#include <iostream>

using namespace std;

int main()

{

int A[3][4]={

{1,-2,3,4},

{3,-4,5,6},

{30,40,50,-60}

};

int first=0;

int second=0;

for(int i=0;i<3;i++){

for(int j=0;j<4;j++){

if(first==0 && A[i][j]<0){

first=10\*i+j;

}

else if(second==0 && A[2-i][3-j]<0 ){

second=10\*(2-i)+3-j;

}

}

}

int counter=0;

for(int i=first/10;i<(second/10);i++){

for(int j=first%10+1;j<4;j++){

counter++;

}

for(int j=0;j<second%10;j++){

counter++;

}

}

cout<<counter;

return 0;

}

11) M(k,l) massivində sətirlər üzrə mütləq qiymətcə elementlərin cəminin maksimum qiymətini tapmalı

#include <iostream>

using namespace std;

int main()

{

int A[3][4]={

{1,-1,3,4},

{3,-4,5,6},

{30,40,50,-60}

};

int sumArray[3];

int index=0;

for(int i=0;i<3;i++){

int sum=0;

for(int j=0;j<4;j++){

sum+=A[i][j];

}

sumArray[index++]=sum;

}

int maxElement=sumArray[0];

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

if(sumArray[j]>maxElement){

maxElement=sumArray[j];

}

}

}

cout<<maxElement;

return 0;

}

12) A(n,n) massivində k-cı sətiri massivdən kənar edib, nəticəni çapa çıxarmalı

#include <iostream>

using namespace std;

int main()

{

int A[4][4]={

{1,-1,3,4},

{3,-4,5,6},

{30,40,50,-60}

};

int newArray[3][4];

int row;

cin>>row;

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

for(int j=0;j<4;j++){

newArray[i][j]=A[i][j];

}

}

for(int i=row+1;i<4;i++){

for(int j=0;j<4;j++){

newArray[i-1][j]=A[i][j];

}

}

for(int i=0;i<3;i++){

for(int j=0;j<4;j++){

cout<<newArray[i][j]<<" ";

}

cout<<endl;

}

return 0;

}