2) M(k,l) matrisində elementlərinin cəmi maksimum olan sətri və sütunu tapıb çapa çıxarmalı

#include <iostream>

using namespace std;

int main()

{

int A[3][4]={

{1,-1,3,4},

{3,-4,5,6},

{30,40,50,-60}

};

int sumArrayRow[4];

int rowIndex=0;

int index=0;

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

int sumRow=0;

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

sumRow+=A[i][j];

}

sumArrayRow[index++]=sumRow;

}

//setr uzre

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

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

if(sumArrayRow[j]>sumArrayRow[rowIndex]){

rowIndex=j;

}

}

}

cout<<"setr uzre"<<endl;

// setr uzre

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

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

}

// sutun uzre

int sumArrayColumn[4];

int columnIndex=0;

index=0;

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

int sumColumn=0;

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

sumColumn+=A[j][i];

}

sumArrayColumn[index++]=sumColumn;

}

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

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

if(sumArrayColumn[j]>sumArrayColumn[columnIndex]){

columnIndex=j;

}

}

}

cout<<"\nsutun uzre"<<endl;

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

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

}

return 0;

}

1) X (n,m) matrisində qiymətlərinin cəmlərinə görə matrisdə minimum olan iki ardıcıl elementi tapıb indekslərini və qiymətlərini çap etməli

#include <iostream>

using namespace std;

int main()

{

int A[3][4]={

{1,-1,3,4},

{3,-4,5,-600},

{30,-140,-50,-60}

};

int index1=-1;

int sumArray[10];

int sum=A[0][0]+A[0][1];

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

int currentSum=0;

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

currentSum=A[i][j]+A[i][j+1];

if(sum>currentSum){

sum=currentSum;

index1=i\*10+j;

}

}

}

cout<<A[index1/10][index1%10]<<" and "<<A[index1/10][index1%10+1];

return 0;

}

3) A(4,5) matrisinin 2-ci sətrindəki elementlərini C ədədinə vur­ma­lı, sonra isə 4-cü sütundakı elementlərini B ədədinə bölməli. Hasil­lərdən və qismətlərdən uyğun olaraq P(5) və T(4) massivlərini təşkil etməli

4) M(k,l) matrisində qiymətlərinin artmasına görə nizamlanmış sətirləri tapıb çapa çıxarmalı, qeyri-nizamlı sətirlərdə isə nizamın pozulduğu 1-ci elementi çapa çıxarmalı

#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[4][4]={

{-2,-2,3,4},

{3,-4,5,-600},

{-300,-140,-50,-60},

{6,7,8,9}

};

int rowArray[4];

int index=0;

bool flag=true;

int element=0;

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

index=0;

flag=true;

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

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

}

bubbleSort(rowArray,4);

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

if(rowArray[j]!=A[i][j]){

element=A[i][j];

cout<<"Pozan ilk element : "<<element<<endl;

flag=false;

break;

}

}

if(flag==true){

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

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

}

cout<<endl;

}

rowArray[4];

}

return 0;

}

5) X (n,m) matrisində özünün bütün qonşularından (sağ, sol, yuxarı və aşağı) böyük olan elementləri tapıb çapa çıxarmalı

#include <iostream>

using namespace std;

int main()

{

int A[4][4]={

{100,2,3,4},

{40,3,4,6},

{30,100,50,60},

{6,7,8,9}

};

int current=0;

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

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

current=A[i][j];

if(current>A[i][j+1]&&i!=0&&j!=0&&current>A[i-1][j]&&current>A[i+1][j]

&&current>A[i][j-1]){

cout<<current<<endl;

}

else if(i==0&&current>A[i][j-1]&&current>A[i][j+1]&&j!=0&&current>A[i+1][j]){

cout<<current<<endl;

}

else if(i!=0&&current>A[i-1][j]&&current>A[i+1][j]&&j==0&&current>A[i][j+1]){

cout<<current<<endl;

}

else if(i==0 &&j==0 &&current>A[i+1][j]&&current>A[i][j+1]){

cout<<current<<endl;

}

}

}

return 0;

}

8) X (n,n) matrisində əvvəlcə yuxarı üçbucağın, sonra aşağı üçbuca­ğın elementlərini çapa çıxarmalı (diaqonal elementlərindən başqa)

#include <iostream>

using namespace std;

int main()

{

int A[4][4]={

{100,2,3,4},

{40,13,14,6},

{30,100,50,60},

{6,7,8,9}

};

int n=4;

cout << "Üst ucbucaq elementler:" << endl;

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

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

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

}

}

cout << "\nAlt ucbucaq elementler:" << endl;

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

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

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

}

}

return 0;

}

9) X (n,m) matrisində yəhər nöqtələrin olub-olmadığını yoxlamalı (yəhər nöqtədəki element öz sətrində minimum və sütununda maksimumdur)

#include <iostream>

using namespace std;

int main()

{

int A[4][4]={

{50,12,31,10},

{40,9,14,1},

{30,10,50,2},

{6,7,8,1}

};

int current=0;

bool flag=true;

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

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

current=A[i][j];

flag=true;

//setr uzre yoxlanis

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

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

flag=false;

break;

}

}

for(int k=j+1;k<4;k++){

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

flag=false;

break;

}

}

//sutun uzre yoxlanish

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

if(current<A[k][j]){

flag=false;

break;

}

}

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

if(current<A[k][j]){

flag=false;

break;

}

}

//cout<<"Element: "<<A[i][j]<<endl;

if(flag==true){

cout<<"Element: "<<A[i][j]<<endl;

}

}

}

return 0;

}

10) X (n,m) matrisində elə üç ardıcıl elementi tapmalı ki, onlar artmayan ardıcıllıq təşkil etsin

int main()

{

int A[4][4]={

{50,12,31,100},

{40,9,14,1},

{30,10,50,2},

{6,7,8,1}

};

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

for(int j = 0; j < 4-2; j++) {

if(A[i][j] <= A[i][j+1] &&( A[i][j+1] <= A[i][j+2] )) {

cout << A[i][j] << " " << A[i][j+1] << " " << A[i][j+2] << endl;

}

}

}

return 0;

}

12) X (10,10) matrisini aşağıdakı şəkildə doldurmalı:



1) #include <iostream>

using namespace std;

int main()

{

int A[10][10];

for (int i = 0; i < 10; i++) {

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

A[i][j]=0;

if(i==j){

A[i][j]=i;

}

}

}

for (int i = 0; i < 10; i++) {

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

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

}

cout<<endl;

}

return 0;

}

2) #include <iostream>

using namespace std;

int main()

{

int A[10][10];

for (int i = 0; i < 10; i++) {

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

A[i][j]=i\*10+1+j;

}

}

for (int i = 0; i < 10; i++) {

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

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

}

cout<<endl;

}

return 0;

}

3) #include <iostream>

using namespace std;

int main()

{

int A[10][10];

int counter=0;

for (int i = 0; i < 10; i++) {

counter=0;

for(int k=i;k<10;k++){

A[i][k]=++counter;

}

}

for (int i = 0; i < 10; i++) {

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

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

}

cout<<endl;

}

return 0;}