RajvaibhavRahane

17u283 223045

SE-C Comp,Viit,Pune

***Aim- To find Upper and Lower Triangular Matrix and Saddle Point***

***CODE:***

#include<iostream>

using namespace std;

typedef struct Matrix{

int \*matrix;

int rows,columns;

}Matrix;

enum TriangularMatrix{FULL=0,UPPER=1,LOWER=2};

Matrix createMatrix(int rows,int columns){

Matrix m;

m.matrix=new int[rows\*columns];

m.rows=rows;

m.columns=columns;

cout<<"Enter Elements:\n";

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

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

cin>>\*((m.matrix+i\*columns) + j);

}

}

return m;

}

void printMatrix(Matrix m){

for(int i=0;i<m.rows;i++){

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

cout<<\*((m.matrix+i\*m.columns)+ j)<<" ";

cout<<endl;

}

}

Matrix getTriangularMatrixOf(Matrix m,TriangularMatrix matrixType){

for(int i=0;i<m.rows;i++){

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

if(matrixType==UPPER){

if(j<i)

\*((m.matrix+i\*m.columns)+j)=0;

}

else if(matrixType==LOWER){

if(i<j)

\*((m.matrix+i\*m.columns)+j)=0;

}

}

}

return m;

}

void findSaddlePoints(Matrix m){

int saddlePoints=0,i,j,k;int minElementInRow,columnIndexOfMinElement;

if(m.rows==m.columns)

{for(i=0;i<m.rows;i++){

minElementInRow=\*((m.matrix+i\*m.columns)+0);columnIndexOfMinElement=0;

for(j=1;j<m.columns;j++){ //find the min element in i'th row

if(minElementInRow>\*((m.matrix+i\*m.columns)+j)){

minElementInRow=\*((m.matrix+i\*m.columns)+j);

columnIndexOfMinElement=j; //set the column of that min element

}

}

for(k=0;k<m.rows;k++){

if(minElementInRow<\*((m.matrix+k\*m.columns)+columnIndexOfMinElement)){

break; //not a saddle point,not a max element in its column

}

}

if(k==m.rows){ //saddle point found

saddlePoints++;

cout<<"Saddle Point Found\n";

cout<<"Value : "<<minElementInRow<<" i : "<<i<<" j : "<<columnIndexOfMinElement<<endl;

}

}

}

cout<<"Saddle Points Found "<<saddlePoints<<endl;

}

/\*void findSaddlePoint(Matrix m){

int minInRow[m.rows]={0},maxInColumn[m.columns]={0};

for(int i=0;i<m.rows<i++){

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

if(\*((m.matrix+i\*m.columns)+j)>maxInColumn[i])

maxInColumn[i]=\*((m.matrix+i\*m.columns)+j);

if(\*((m.matrix+j\*m.columns)+i)<minInRow[i])

minInRow[i]

}

}

}\*/

int main(){

int r,c;

cin>>r>>c;

Matrix matrix=createMatrix(r,c);

findSaddlePoints(matrix);

if(r==c){

cout<<"\nSquare Matrix\n";

Matrix utMatrix=getTriangularMatrixOf(matrix,UPPER);

printMatrix(utMatrix);cout<<endl;

Matrix ltMatrix=getTriangularMatrixOf(matrix,LOWER);

printMatrix(ltMatrix);

}

else{

cout<<"Not a Sqare Matrix";

}

return 0;

}

***Output:***

