**K.Kalyani**

**21071A6793**

**1 .Given a row wise sorted matrix of size R\*C where R and** **C** **are always odd, find the median of the matrix.**

#include<bits/stdc++.h>

using namespace std;

int median(int a[][100],int row,int column)

{

int n=row\*column;

int array[n],index=0,i,j;

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

{

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

{

array[index]=a[i][j];

index++;

}

}

sort(array,array+n);

int mid=n/2;

if(n%2==1)

return array[n/2];

else

return array[n/2]+array[n/2+1];

}

int main()

{

int r,c,i,j;

cin>>r>>c;

cout<<"Array elements are";

int a[100][100];

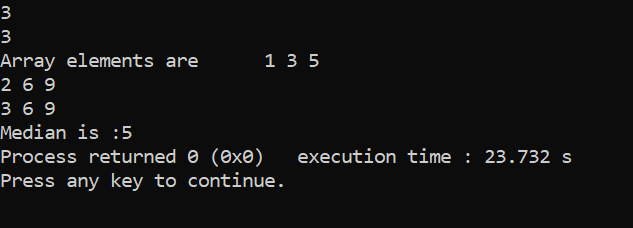
for(i=0;i<r;i++)

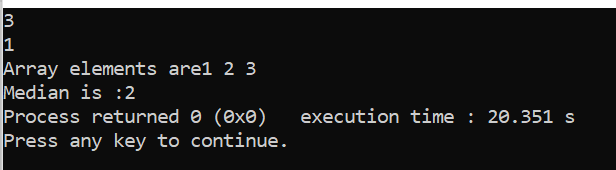
for(j=0;j<c;j++)

cin>>a[i][j];

cout<<"Median is :"<<median(a,r,c);

return 0;

}



**2. Given the arrival and departure times of all trains that reach a railway station, the task is to find the minimum number of platforms required for the railway station so that no train waits. We are given two arrays that represent the arrival and departure times of trains that stop.**

#include <bits/stdc++.h>

using namespace std;

int ans(int arr[],int dep[],int n)

{

int min=0,i,j;

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

{

int p=1;

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

{

if ((arr[i]>=arr[j] && arr[i]<= dep[j]) || (arr[j]>=arr[i] && arr[j]<=dep[i]))

p++;

}

min = max(min,p);

} return min;

}

int main()

{

int n,i;

cin>>n;

int arr[n],dep[n];

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

cin>>arr[i];

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

cin>>dep[i];

cout <<ans(arr,dep,n);

return 0;

}



