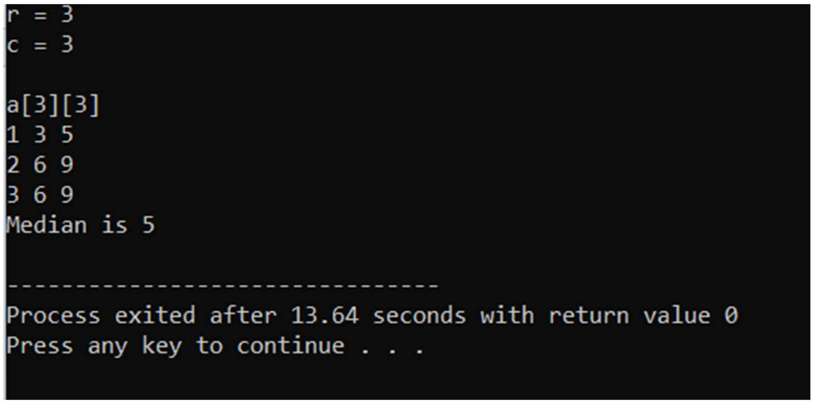
DAA ASSIGNMENT-21071A67C8

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

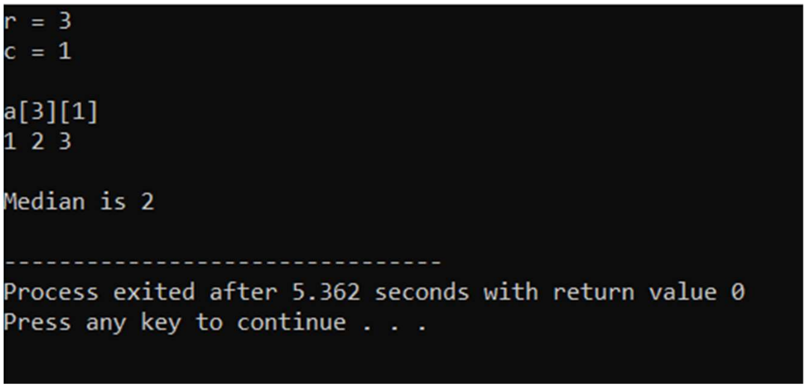
```
#include<bits/stdc++.h>
using namespace std;
const int MAX = 100;
int bm(int a[][MAX], int r ,int c)
{
int min = INT_MAX, max = INT_MIN;
        for (int i=0; i<r; i++)
          {
                  if (a[i][0] < min)
                    min = a[i][0];
                 if (a[i][c-1] > max)
                     max = a[i][c-1];
          }
         int x= (r * c + 1) / 2;
          while (min < max)
          {
                  int mid = min + (max - min) / 2;
                    int p = 0;
                 for (int i = 0; i < r; ++i)
                             p += upper_bound(a[i], a[i]+c, mid) - a[i];
                    if (p < x)
                             min = mid + 1;
                    else
                          max = mid;
          }
          return min;
}
int main()
{
   int r,c,a[MAX][MAX];

  cout<<"r = ";

  cin>>r;

  cout<<"c = ";

  cin>>c;

  cout<<"\na["<<r<<"]["<<c<<"]\n";

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

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

     cin>>a[i][j];

cout << "\nMedian is "<< bm(a, r, c) << endl;

return 0;

}
```

TESTCASE 1:



```
r = 3
c = 3

a[3][3]
1 3 5
2 6 9
3 6 9
Median is 5

--------------------------------
Process exited after 13.64 seconds with return value 0
Press any key to continue . . .
```

TESTCASE2:



```
r = 3
c = 1

a[3][1]
1 2 3

Median is 2

--------------------------------
Process exited after 5.362 seconds with return value 0
Press any key to continue . . .
```

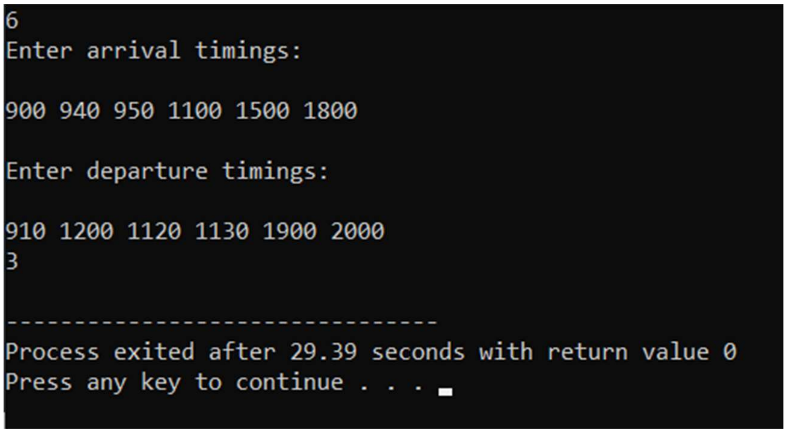
```
}
```

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 stops.

CODE:

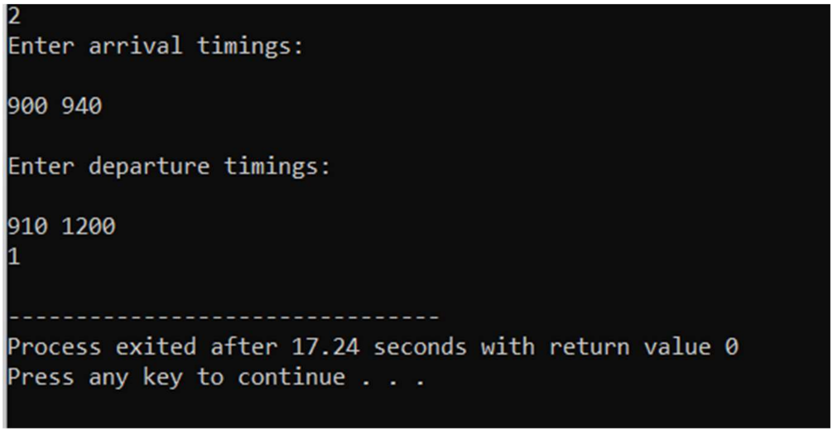
```
#include<bits/stdc++.h>
using namespace std;
int main()
{
 int n;
 cin>>n;
 int arrival[n],departure[n];
 cout<<"Enter arrival timings:\n"<<endl;
 for(int i=0;i<n;i++)
 cin>>arrival[i];
 cout<<"Enter departure timings:\n"<<endl;
 for(int i=0;i<n;i++)
 cin>>departure[i];
 sort(arrival,arrival+n);
 sort(departure,departure+n);
 int platforms_needed=1,result=1,i=1,j=0;
while (i<n&&j<n){
  if(arrival[i]<=departure[j])
  {
   platforms_needed++;
   i++;
  }
  else if(arrival[i]>departure[j]){
   platforms_needed--;
   j++;
  }
 if(platforms_needed > result)
 result=platforms_needed;
 }
 cout<<result;
 return 0;
}
```

TESTCASE 1:



```
6
Enter arrival timings:

900 940 950 1100 1500 1800

Enter departure timings:

910 1200 1120 1130 1900 2000
3

--------------------------------
Process exited after 29.39 seconds with return value 0
Press any key to continue . . .
```

TESTCASE 2:



```
2
Enter arrival timings:

900 940

Enter departure timings:

910 1200
1

--------------------------------
Process exited after 17.24 seconds with return value 0
Press any key to continue . . .
```