

DAA ASSIGNMENT

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

main.cpp

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  class median{
4      private:
5          int m,n;
6          void set(){
7              cin>>m>>n;
8              int p=m*n;
9              int a[p],i;
10             cout<<"Enter elements in the array"<<endl;
11             for(i=0;i<p;i++){
12                 cin>>a[i];
13             }
14             sort(a,a+p);
15             p=(p-1)/2;
16             cout<<"Median is: "<<a[p];
17         }
18     public:
19         median(){
20             set();
21         }
22 };
23 int main() {
24     median k;
25     return 0;
26 }
```

Input:

R = 3, C = 3

M = [[1, 3, 5],
 [2, 6, 9],
 [3, 6, 9]]

Output: 5

Explanation: Sorting matrix elements gives us {1,2,3,3,5,6,6,9,9}. Hence, 5 is median.

Output

/tmp/5FdqE7q4C8.o

3

3

Enter elements in the array

1

3

5

2

6

9

3

6

9

Median is: 5

Input:

R = 3, C = 1

M = [[1], [2], [3]]

Output: 2

Explanation: Sorting matrix elements gives us {1,2,3}. Hence, 2 is median.

Output

/tmp/5FdqE7q4C8.o

3

1

Enter elements in the array

1

2

3

Median is: 2

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.

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  class platform{
4      private:
5          int n,i,arr[100],dep[100];
6          void set(){
7              cout<<"enter number of trains"<<endl;
8              cin>>n;
9              cout<<"enter arrival times"<<endl;
10             for(i=0;i<n;i++)
11                 cin>>arr[i];
12             sort(arr,arr+n);
13             cout<<"enter departure times"<<endl;
14             for(i=0;i<n;i++)
15                 cin>>dep[i];
16             sort(dep,dep+n);
17         }
18     public:
19         platform(){
20             set();
21             int i=1,j=0,p=1;
22             while(i<n&& j<n){
23                 if(arr[i]>dep[j])
24                     j++;
25                 else
26                     p++;
27                 i++;
28             }
29             cout<<"number of platforms : "<<p;
30         }
31     };
32 int main() {
33     platform q;
34     return 0;
35 }
```

Test case 1:

Input: arr[] = {9:00, 9:40, 9:50, 11:00, 15:00, 18:00}, dep[] = {9:10, 12:00, 11:20, 11:30, 19:00, 20:00}

Output: 3

Explanation: There are at-most three trains at a time (time between 9:40 to 12:00)

```
Output
/tmp/5FdqE7q4C8.o
enter number of trains
6
enter arrival times
900
940
950
1100
1500
1800
enter departure times
910
1200
1120
1130
1900
2000
number of platforms : 3
```

Test case 2:

Input: arr[] = {9:00, 9:40}, dep[] = {9:10, 12:00}

Output: 1

Explanation: Only one platform is needed.

```
Output
/tmp/5FdqE7q4C8.o
enter number of trains
2
enter arrival times
900
940
enter departure times
910
1200
number of platforms : 1
```