

DAA Assignment -1

- 21071A6769

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. 5Marks

Test Case 1:

```
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.
```

Test Case 2:

```
Input:
R = 3, C = 1
M = [[1], [2], [3]]
Output: 2
Explanation: Sorting matrix elements gives us
{1,2,3}. Hence, 2 is median.
```

● Constraints:

$1 \leq R, C \leq 400$

$1 \leq \text{matrix}[i][j] \leq 2000$

Code(python):

```

1 row=int(input('Enter number of rows: '))
2
3 col=int(input('Enter number of columns: '
               ))
4
5 m=[]
6 print('Enter matrix elements: ')
7 #taking space separated integers as input
  for each row separately
8 for i in range(row):
9     a=list(map(int,input().split()))
10    m.append(a)
11 import numpy
12 median=numpy.median(m)
13 print(f'Median of elements of matrix is:
        {median}') |

```

```

OUTPUT - 1
Enter number of rows: 3
Enter number of columns: 3
Enter matrix elements:
1 3 5
2 6 9
3 6 9
Median of elements of matrix is: 5.0

```

```

Enter number of rows: 3
Enter number of columns: 1
Enter matrix elements:
1
2
3
Median of elements of matrix is: 2.0
>

```

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.

5Marks

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)

Test case 2

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

Output: 1

Explanation: Only one platform is needed.

Code(python):

```
arrive=list(input('Enter arrival timings of trains: ').split())
dep=list(input('Enter departure timings of trains: ').split())

arrival=[]
departure=[]

for i in arrive:
    i=list(i)
    i.remove(':')
    j="".join(i)
    j=int(j)
    arrival.append(j)

for i in dep:
    i=list(i)
    i.remove(':')
    j="".join(i)
    j=int(j)
    departure.append(j)

n=len(arrival)
arrival.sort()
departure.sort()

platforms=1
req_platforms=1
i,j=1,0

while i<n and j<n:
    if arrival[i]<=departure[j]:
        platforms+=1
        i+=1
    else:
        j+=1
```

```
platforms-=1  
j+=1  
req_platforms=max(req_platforms,platforms)  
print(f'Required number of platforms is: {req_platforms}')
```

#end of program

Output-1:

```
Enter arrival timings of trains: 9:00 9:40 9:50 11:00 15:00 18:00  
Enter departure timings of trains: 9:10 12:00 11:20 11:30 19:00 20:00  
Required number of platforms is: 3  
>
```

Output-2:

```
Enter arrival timings of trains: 9:00 9:40  
Enter departure timings of trains: 9:10 12:00  
Required number of platforms is: 1  
> |
```