

21071A67A8

DAA ASSIGNMENT

QUESTION 1:

QUESTION

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.

CODE:

```
arr=list(map(int,input().split()))
dep=list(map(int,input().split()))
l=len(arr)
arr.sort()
dep.sort()
x,y=1,1
i,j=1,0
while i<l and j<l:
    if arr[i]<=dep[j]:
        x=x+1
```

```
i=i+1  
else:  
    x=x-1  
    j=j+1  
y=max(x,y)  
print(y)
```

```
main.py  
1 R,C=map(int,input().split(' '))  
2 M=[]  
3 for i in range(R):  
4     A=[]  
5     for j in range(C):  
6         A.append(int(input()))  
7     M.append(A)  
8 M.sort()  
9 import numpy as nk  
10 X=nk.median(M)  
11 print(X)
```

OUTPUT 1:

```
input  
3 3  
1  
3  
5  
2  
6  
9  
3  
6  
9  
5.0
```

OUTPUT 2:

```
input
3 1
1
2
3
2.0

...Program finished with exit code 0
Press ENTER to exit console.
```

QUESTION 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.

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:

```
R,C=(int,input().split(' '))
```

```
M=[]
```

```
for i in range(R):
```

```
    A=[]
```

```
for j in range(C):  
    A.append(int(input()))  
  
M.append(a)  
  
B=sort(M)  
  
import numpy as nk  
  
X=nk.median(B)  
  
print(X)
```

```
1 arr=list(map(int,input().split()))  
2 dep=list(map(int,input().split()))  
3 l=len(arr)  
4 arr.sort()  
5 dep.sort()  
6 x,y=1,1  
7 i,j=1,0  
8 while i<l and j<l:  
9     if arr[i]<=dep[j]:  
10         x=x+1  
11         i=i+1  
12     else:  
13         x=x-1  
14         j=j+1  
15     y=max(x,y)  
16 print(y)
```

OUTPUT:

```
900 940 950 1100 1500 1800  
910 1200 1120 1130 1900 2000  
3  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```



```
900 940
910 1200
1

...Program finished with exit code 0
Press ENTER to exit console.
```