DAA Assignment -1

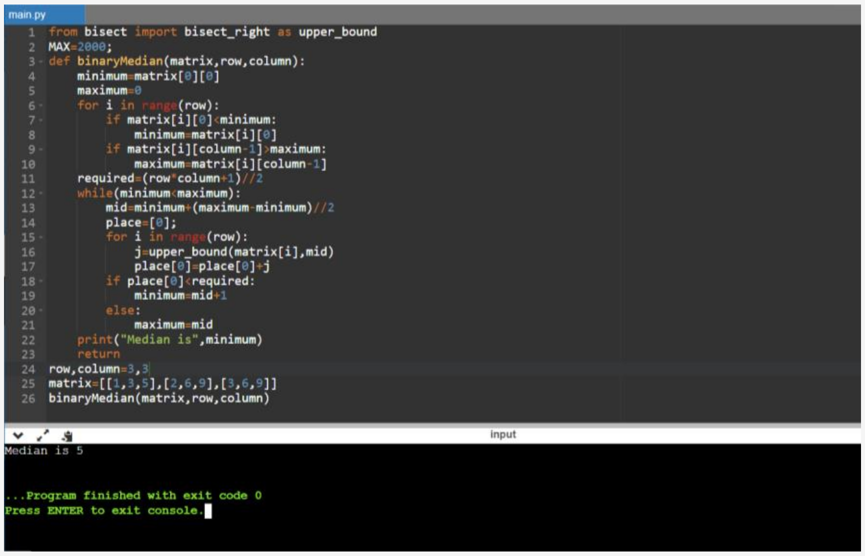
(Implements the following problems using C++ / Python)

21071A6785-G.VARUN RAJ-CSDS-B

1 .Given a row wise sorted matrix of size **R*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.
```



main.py

```
from bisect import bisect_right as upper_bound
MAX=2000;
def binaryMedian(matrix,row,column):
    minimum=matrix[0][0]
    maximum=0
    for i in range(row):
        if matrix[i][0]<minimum:
            minimum=matrix[i][0]
        if matrix[i][column-1]>maximum:
            maximum=matrix[i][column-1]
    required=(row*column+1)//2
    while(minimum<maximum):
        mid=minimum+(maximum-minimum)//2
        place=[0];
        for i in range(row):
            j=upper_bound(matrix[i],mid)
            place[0]=place[0]+j
        if place[0]<required:
            minimum=mid+1
        else:
            maximum=mid
    print("Median is",minimum)
    return
row,column=3,3
matrix=[[1,3,5],[2,6,9],[3,6,9]]
binaryMedian(matrix,row,column)
```

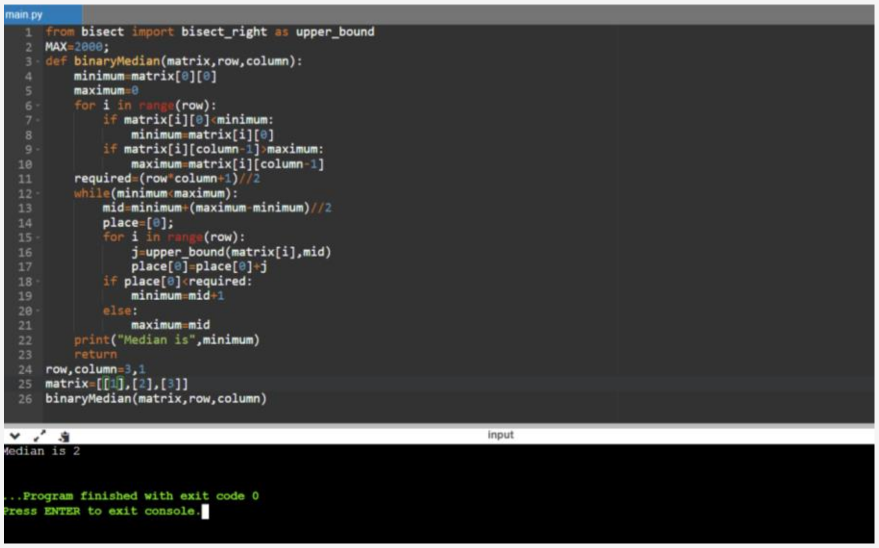
```
Median is 5


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

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.


main.py

```python
from bisect import bisect_right as upper_bound
MAX=2000;
def binaryMedian(matrix,row,column):
    minimum=matrix[0][0]
    maximum=0
    for i in range(row):
        if matrix[i][0]<minimum:
            minimum=matrix[i][0]
        if matrix[i][column-1]>maximum:
            maximum=matrix[i][column-1]
    required=(row*column+1)//2
    while(minimum<maximum):
        mid=minimum+(maximum-minimum)//2
        place=[0];
        for i in range(row):
            j=upper_bound(matrix[i],mid)
            place[0]=place[0]+j
        if place[0]<required:
            minimum=mid+1
        else:
            maximum=mid
    print("Median is",minimum)
    return
row,column=3,1
matrix=[[1],[2],[3]]
binaryMedian(matrix,row,column)
```

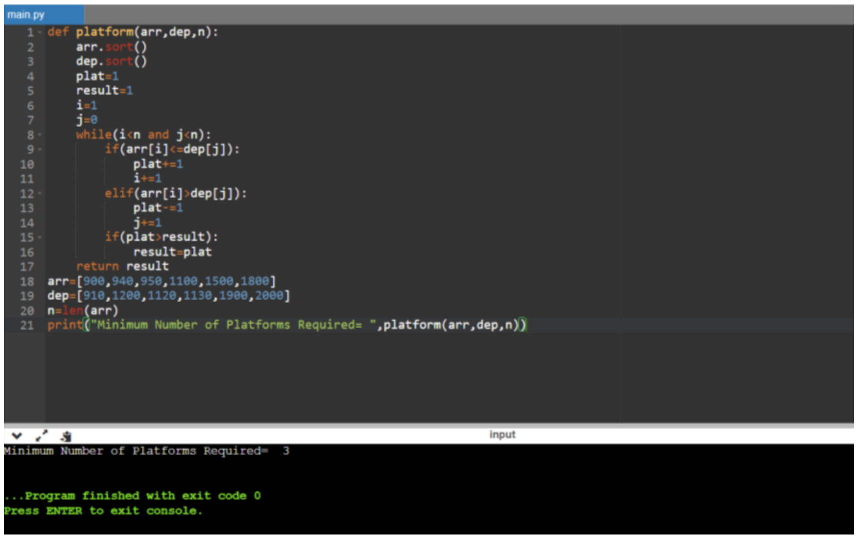
```
Median is 2


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

➤ **Constraints:**
1 <= R, C <= 400
1 <= matrix[i][j] <= 2000

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)*



```python
def platform(arr,dep,n):
    arr.sort()
    dep.sort()
    plat=1
    result=1
    i=1
    j=0
    while(i<n and j<n):
        if(arr[i]<=dep[j]):
            plat+=1
            i+=1
        elif(arr[i]>dep[j]):
            plat-=1
            j+=1
        if(plat>result):
            result=plat
    return result
arr=[900,940,950,1100,1500,1800]
dep=[910,1200,1120,1130,1900,2000]
n=len(arr)
print("Minimum Number of Platforms Required= ",platform(arr,dep,n))
```

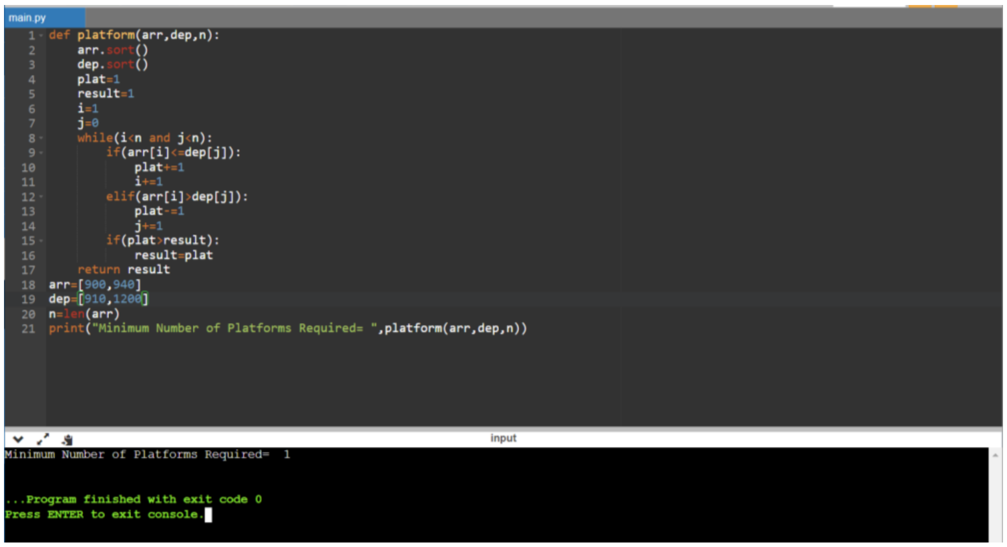
```
Minimum Number of Platforms Required=  3


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

Test case 2

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

Output: *1*

Explanation: *Only one platform is needed.*



```python
def platform(arr,dep,n):
    arr.sort()
    dep.sort()
    plat=1
    result=1
    i=1
    j=0
    while(i<n and j<n):
        if(arr[i]<=dep[j]):
            plat+=1
            i+=1
        elif(arr[i]>dep[j]):
            plat-=1
            j+=1
        if(plat>result):
            result=plat
    return result
arr=[900,940]
dep=[910,1200]
n=len(arr)
print("Minimum Number of Platforms Required= ",platform(arr,dep,n))
```

```
Minimum Number of Platforms Required=  1


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