NAME : B. Harshini

Hall Ticket.No: 2303A52242

Batch:36

Assignment 1: Maximum Non-Overlapping Meetings (Greedy)

Problem Statement

You are given N meetings. Each meeting has a start time Si and an end time Ei. You want to attend the maximum number of meetings. You can attend meeting j after meeting i only if the start time of meeting j is strictly greater than the end time of meeting i (Sj > Ei). For each test case, output the maximum number of meetings that can be attended.

Input Format

The first line contains an integer T, the number of test cases.For each test case:

- The first line contains an integer N.

- The next N lines each contain two integers Si and Ei.

Output Format: For each test case, print a single integer: the maximum number of meetings that can be attended.

Constraints

- $1 \leq T \leq 20$

- $1 \leq N \leq 200000$ (sum of N over all test cases ≤ 200000)

- $0 \leq Si < Ei \leq 10^9$

Sample Input

1

3

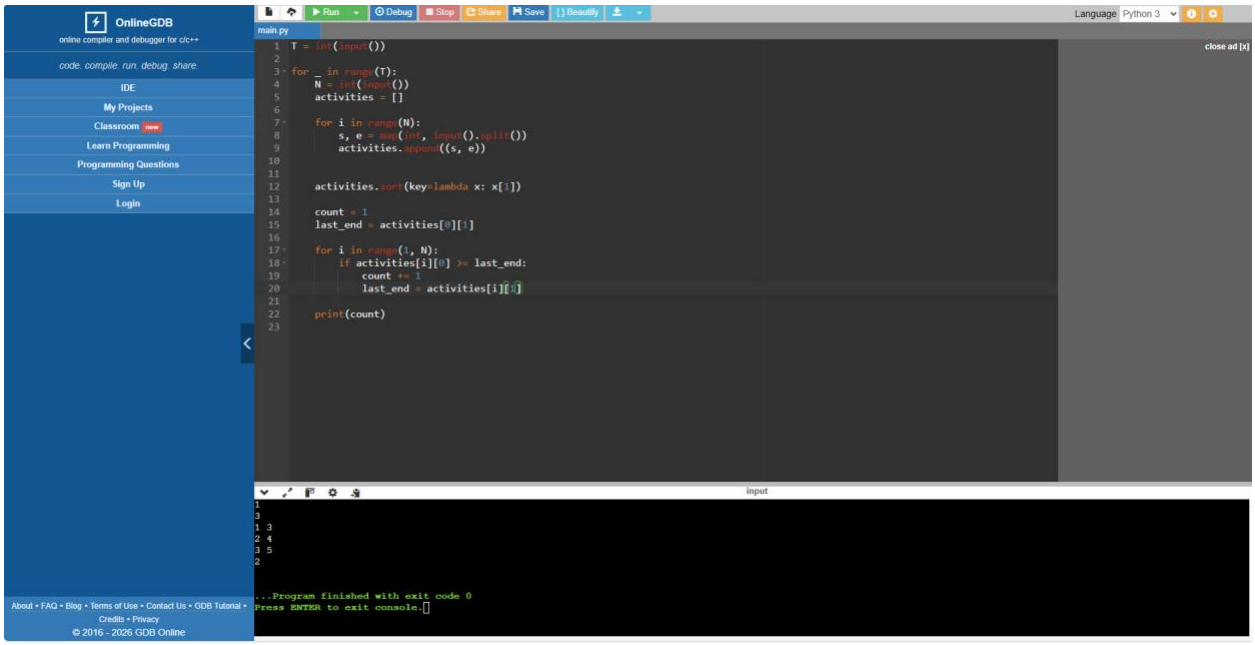
1 3

2 4

3 5

Expected Output : 2

CODE: (PYTHON)



CODE: (JAVA)

Assignment-1: Shelf Product Lookup (Divide and Conquer)

Problem Statement

A supermarket maintains a sorted list of product IDs. For each query ID X, determine how many times X appears in the list. You must solve using binary-search style divide and conquer (find first occurrence and last occurrence).

Input FormatThe first line contains integer T. For each test case:

- First line: N Q

- Second line: N integers (sorted product IDs)

- Next Q lines: integer X (query ID)

Output FormatFor each query, print the count of X on a new line.

Constraints

- $1 \leq T \leq 20$

- $1 \leq N \leq 200000$ (sum of N over all test cases ≤ 200000)

- $1 \leq Q \leq 200000$ (sum of Q over all test cases ≤ 200000)

- Product IDs fit in 32-bit signed integer

Sample Input

1

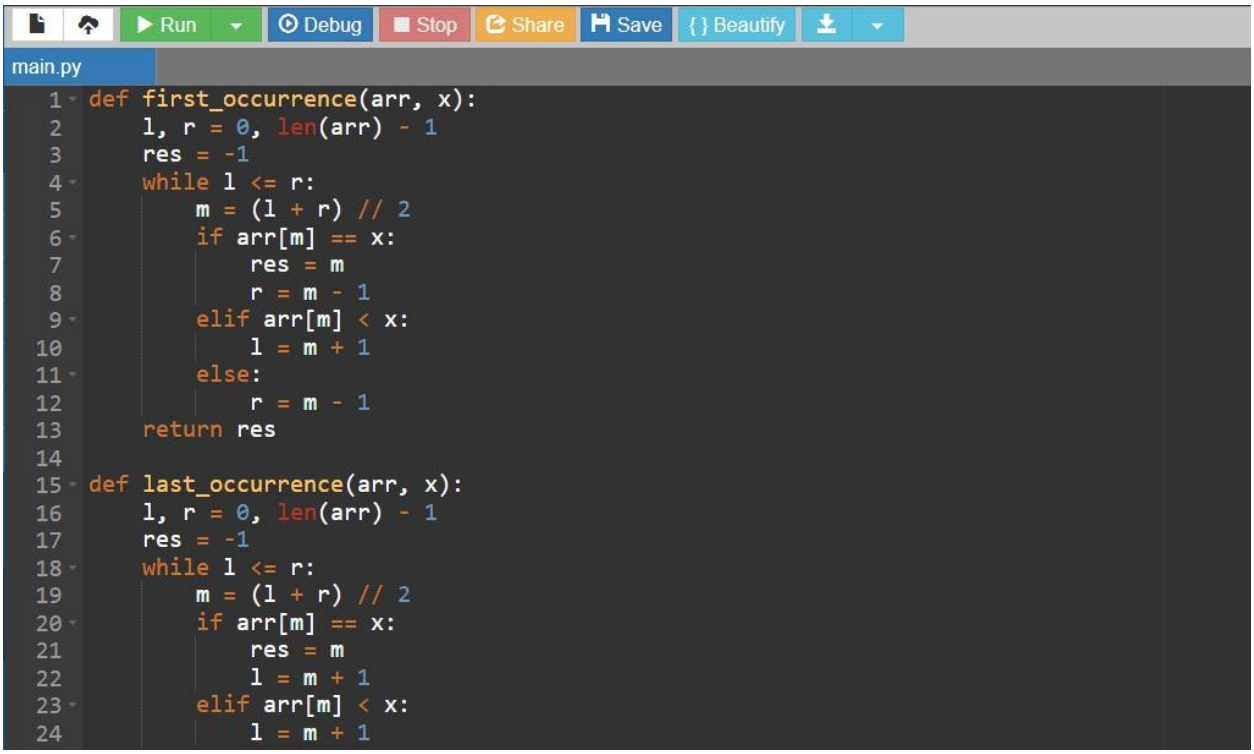
8 3

101 101 102 102 102 105 110 110 101 102 999

Expected Output

230

Python code:

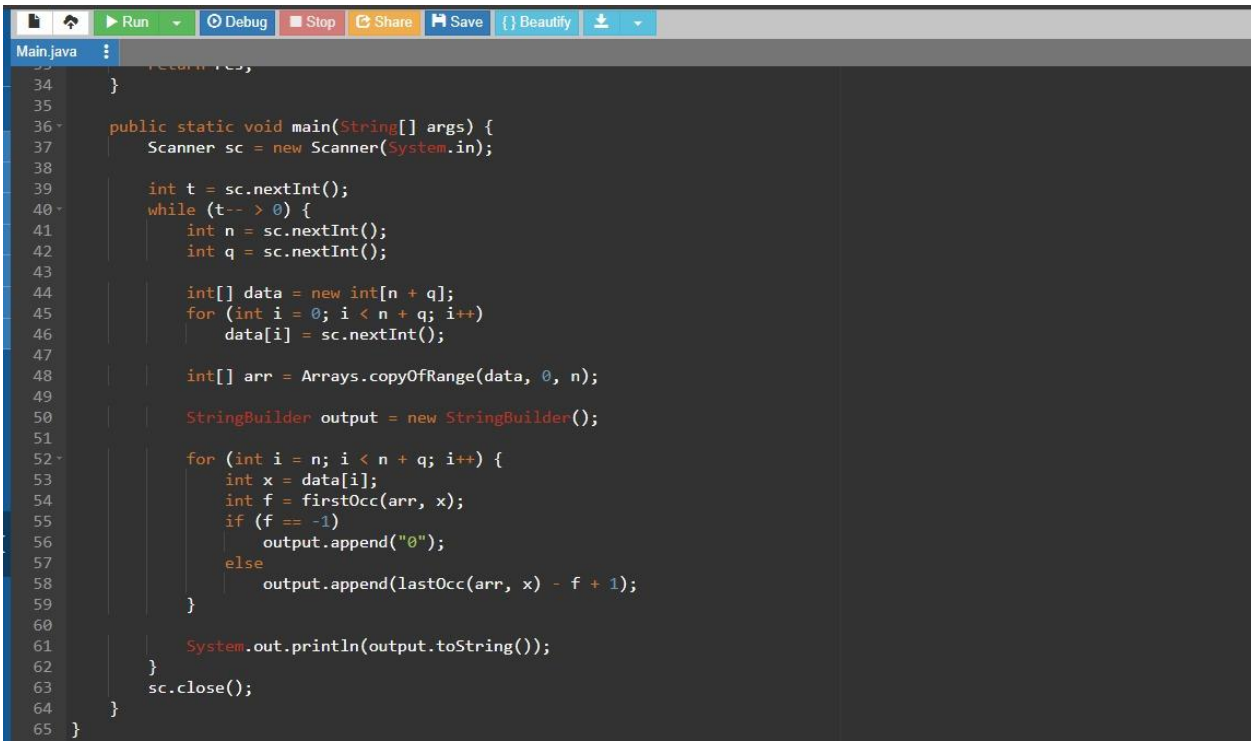


```
            r = m - 1
        elif arr[m] < x:
            l = m + 1
        else:
            r = m - 1
    return res
def last_occurrence(arr, x):
    l, r = 0, len(arr) - 1
    res = -1
    while l <= r:
        m = (l + r) // 2
        if arr[m] == x:
            res = m
            l = m + 1
        elif arr[m] < x:
            l = m + 1
        else:
            r = m - 1
    return res
t = int(input())
for _ in range(t):
    n, q = map(int, input().split())
    data = list(map(int, input().split()))

    arr = data[:n]          # product IDs
    queries = data[n:n+q]   # query IDs

    result = ""
    for x in queries:
        f = first_occurrence(arr, x)
        if f == -1:
            result += "0"
        else:
            l = last_occurrence(arr, x)
            result += str(l - f + 1)
    print(result)
```

Output:

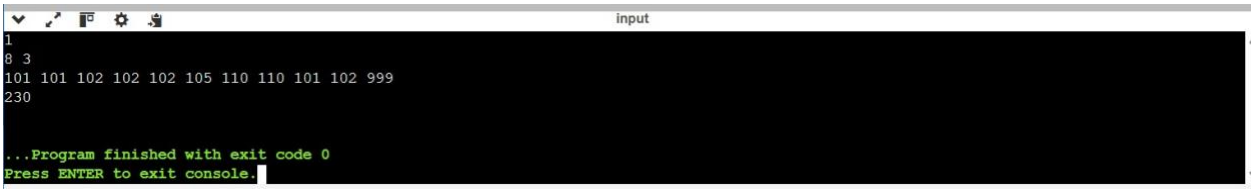
Java Code:

```java
import java.util.*;
class Main {

    static int firstOcc(int[] arr, int x) {
        int l = 0, r = arr.length - 1, res = -1;
        while (l <= r) {
            int m = (l + r) / 2;
            if (arr[m] == x) {
                res = m;
                r = m - 1;
            } else if (arr[m] < x) {
                l = m + 1;
            } else {
                r = m - 1;
            }
        }
        return res;
    }

    static int lastOcc(int[] arr, int x) {
        int l = 0, r = arr.length - 1, res = -1;
        while (l <= r) {
            int m = (l + r) / 2;
            if (arr[m] == x) {
                res = m;
                l = m + 1;
            } else if (arr[m] < x) {
                l = m + 1;
            } else {
                r = m - 1;
            }
        }
        return res;
```



```java
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int t = sc.nextInt();
        while (t-- > 0) {
            int n = sc.nextInt();
            int q = sc.nextInt();

            int[] data = new int[n + q];
            for (int i = 0; i < n + q; i++)
                data[i] = sc.nextInt();

            int[] arr = Arrays.copyOfRange(data, 0, n);

            StringBuilder output = new StringBuilder();

            for (int i = n; i < n + q; i++) {
                int x = data[i];
                int f = firstOcc(arr, x);
                if (f == -1)
                    output.append("0");
                else
                    output.append(lastOcc(arr, x) - f + 1);
            }

            System.out.println(output.toString());
        }
        sc.close();
    }
}
```

Output:


input
1
8 3
101 101 102 102 102 105 110 110 101 102 999
230
...Program finished with exit code 0
Press ENTER to exit console.