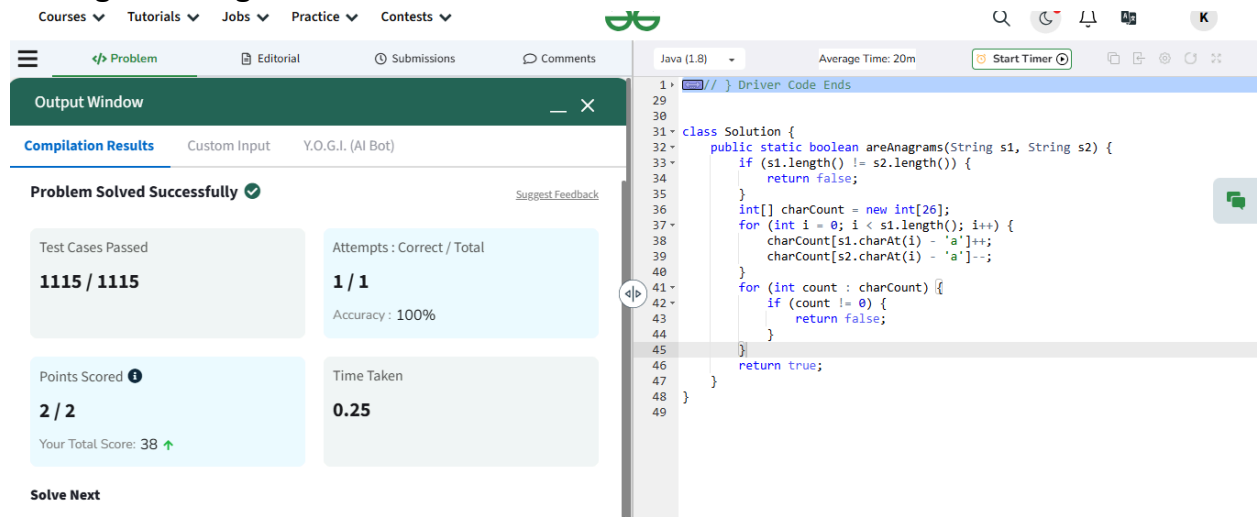


# DSA Coding Practice-3

## 1. Anagram Program:



**Output Window**

**Compilation Results** Custom Input Y.O.G.I. (AI Bot)

**Problem Solved Successfully** ✓ [Suggest Feedback](#)

Test Cases Passed: **1115 / 1115**

Attempts: Correct / Total: **1 / 1**  
Accuracy: 100%

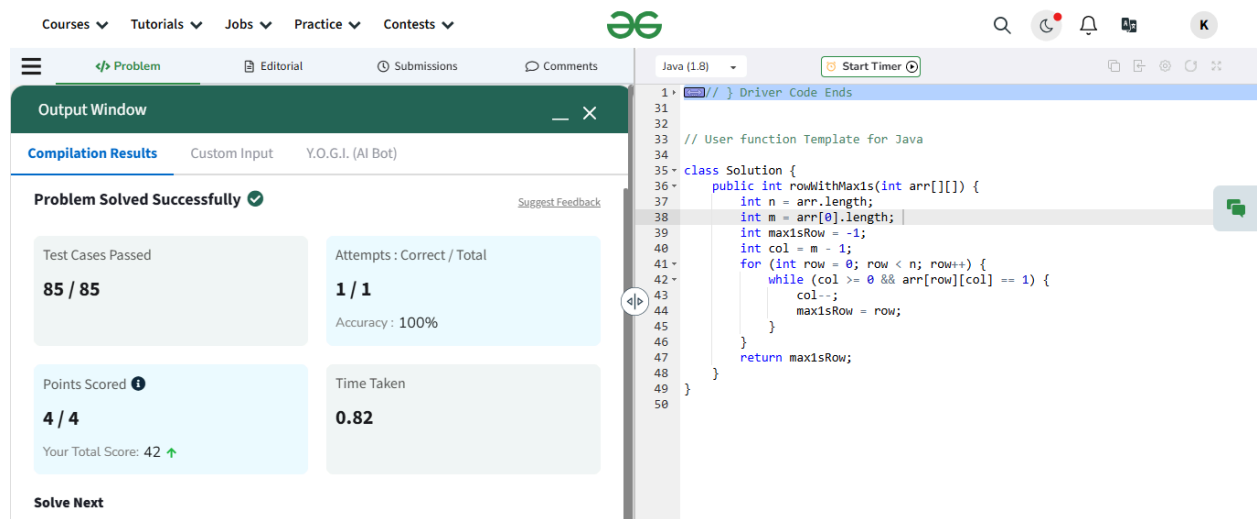
Points Scored: **2 / 2**  
Your Total Score: 38 ↑

Time Taken: **0.25**

**Solve Next**

```
1 // Driver Code Ends
29
30
31 class Solution {
32     public static boolean areAnagrams(String s1, String s2) {
33         if (s1.length() != s2.length()) {
34             return false;
35         }
36         int[] charCount = new int[26];
37         for (int i = 0; i < s1.length(); i++) {
38             charCount[s1.charAt(i) - 'a']++;
39             charCount[s2.charAt(i) - 'a']--;
40         }
41         for (int count : charCount) {
42             if (count != 0) {
43                 return false;
44             }
45         }
46         return true;
47     }
48 }
49
```

## 2. Row with max 1s:



**Output Window**

**Compilation Results** Custom Input Y.O.G.I. (AI Bot)

**Problem Solved Successfully** ✓ [Suggest Feedback](#)

Test Cases Passed: **85 / 85**

Attempts: Correct / Total: **1 / 1**  
Accuracy: 100%

Points Scored: **4 / 4**  
Your Total Score: 42 ↑

Time Taken: **0.82**

**Solve Next**

```
1 // Driver Code Ends
31
32
33 // User function Template for Java
34
35 class Solution {
36     public int rowWithMax1s(int arr[][] ) {
37         int n = arr.length;
38         int m = arr[0].length;
39         int max1sRow = -1;
40         int col = m - 1;
41         for (int row = 0; row < n; row++) {
42             while (col >= 0 && arr[row][col] == 1) {
43                 col--;
44                 max1sRow = row;
45             }
46         }
47         return max1sRow;
48     }
49 }
50
```