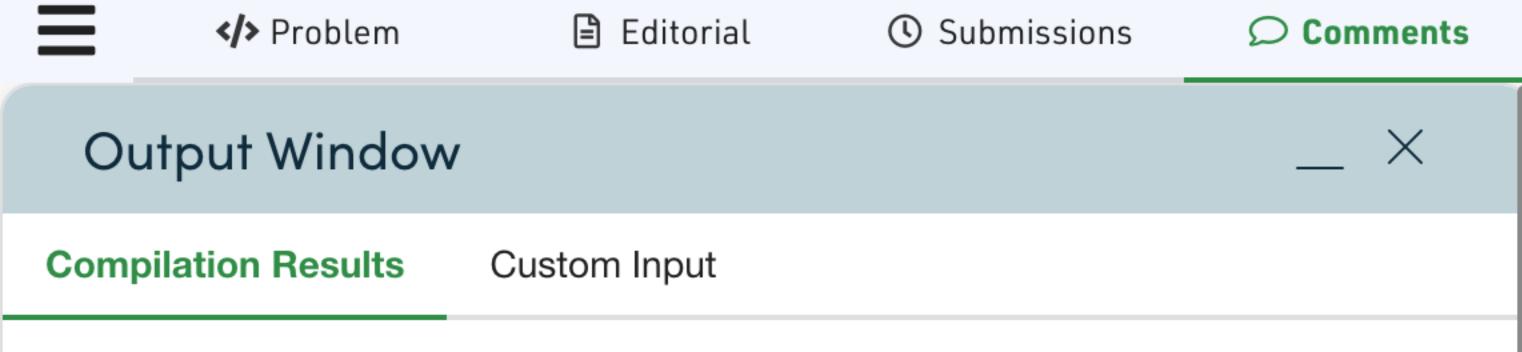
```
//{ Driver Code Starts
#include <iostream>
#include <vector>
using namespace std;
class Solution
public:
    // Function to find length of longest increasing subsequence:
    int lis(vector<int> const &a)
        int n = a.size();
        /*
        Initialise the integer vector d[] with ones,
        used to store the maximum subsequence lengths,
        starting from the first element a[0] and ending
        ending at the ith element a[i]:
        */
        vector<int> d(n, 1);
        // Recursively consider each element of the vector a:
        // For each a[i]:
        for (int i = 0; i < n; i++)
            /*
            Check if each previous element a[j], j < i,
            is strictly less than a[i]:
            for (int j = 0; j < i; j++)
            {
                If a[j] < a[i], store d[i] as the maximum of either:</pre>
                d[i] (if a[j] = a[i]), or d[j] + 1 (if a[j] < a[i]):
                */
                if (a[j] < a[i])
                    d[i] = max(d[i], d[j] + 1);
            }
        }
        int ans = d[0];
        for (int i = 1; i < n; i++)
            ans = max(ans, d[i]);
        return ans;
    }
    // Function to find length of longest increasing subsequence.
    int longestSubsequence(int n, int a[])
    {
```

```
// your code here
         vector<int> vec = {};
         /*
         Convert the integer array a[] into an integer vector vec[], so that we can apply the "lis" function and return the
         length of the largest increasing subsequence:
         for (int i = 0; i < n; i++)
              vec.push_back(a[i]);
         }
         /*
         Calculate the length of the largest increasing subsequence
of
         the array a[], by passing it to the function "lis" (as a
vector):
         */
         int length = lis(vec);
         return length;
    }
};
```



Problem Solved Successfully

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

Test Cases Passed:

111/111

Problems

Your Total Score:

Total Time Taken:

Correct Submission Count:

Attempts No.:

```
class Solution
 9 +
10
    public:
12
        // Function to find length of longest increasing subsequence:
13
        int lis(vector<int> const &a)
14 -
15
            int n = a.size();
16
17 -
            Initialise the integer vector d[] with ones,
19
            used to store the maximum subsequence lengths,
20
            starting from the first element a[0] and ending
            ending at the ith element a[i]:
22
            vector<int> d(n, 1);
24
25
            // Recursively consider each element of the vector a:
26
            // For each a[i]:
27
            for (int i = 0; i < n; i++)
28 -
29 -
30
                Check if each previous element a[j], j < i,
31
                is strictly less than a[i]:
32
                */
33
                for (int j = 0; j < i; j++)
34 -
35 -
36
                    If a[j] < a[i], store d[i] as the maximum of either:
37
                    d[i] (if a[j] = a[i]), or d[j] + 1 (if a[j] < a[i]):
30
  -<u>;</u>Ö́;-
```