Started on	Wednesday, 20 November 2024, 8:25 AM
State	Finished
Completed on	Wednesday, 20 November 2024, 8:26 AM
Time taken	1 min 2 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Eg:

Input:9

Sequence:[-1,3,4,5,2,2,2,2,3]

the subsequence is [-1,2,2,2,2,3]

Output:6

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

// Function to find the length of the Longest Non-decreasing Subsequence
int longestNonDecreasingSubsequence(int arr[], int n) {
    int dp[n]; // dp[i] represents the length of LNDS ending at index i
    int maxLength = 1; // At least one element is a subsequence

    // Initialize dp array to 1 since each element is a subsequence of length 1
    for (int i = 0; i < n; i++) {
        dp[i] = 1;
    }

    // Build dp array
    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (arr[i] >= arr[j] && dp[i] < dp[j] + 1) {
                dp[i] = dp[j] + 1;
            }
        }
        // Update maximum length
        if (dp[i] > maxLength) {
            maxLength = dp[i];
        }
    }

    return maxLength;
}

int main() {
    int n;

    // Input the size of the sequence
    scanf("%d", &n);
    int arr[n];

    // Input the sequence
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Find and print the length of the Longest Non-decreasing Subsequence
    printf("%d\n", longestNonDecreasingSubsequence(arr, n));

    return 0;
}

```

	Input	Expected	Got	
✔	9 -1 3 4 5 2 2 2 2 3	6	6	✔

	Input	Expected	Got	
✔	7 1 2 2 4 5 7 6	6	6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 3-DP-Longest Common Subsequence

Jump to...

1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity ►