Dynamic Programming

Program – 4

Aim:

To find the length of the Longest Non-decreasing Subsequence (LNDS) in a given sequence.

Input:

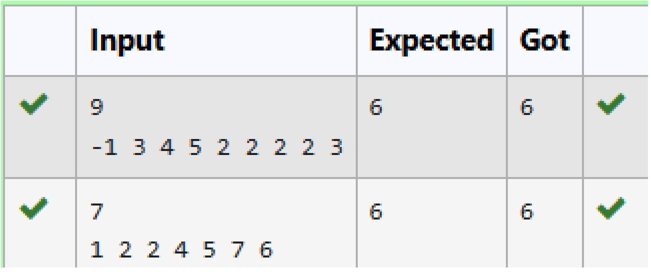
- The first line contains the integer nnn (size of the sequence).
- The second line contains the sequence of integers.

Code:

```
#include <stdio.h>


int longest_non_decreasing_subsequence(int* sequence, int n) {

   int dp[n];


   for (int i = 0; i < n; i++) {

       dp[i] = 1;

   }


   for (int i = 1; i < n; i++) {

       for (int j = 0; j < i; j++) {

           if (sequence[i] >= sequence[j]) {

               dp[i] = (dp[i] > dp[j] + 1) ? dp[i] : dp[j] + 1;

           }

       }

   }
```

```
    int max_length = dp[0];
    for (int i = 1; i < n; i++) {
        if (dp[i] > max_length) {
            max_length = dp[i];
        }
    }

    return max_length;
}

int main() {
    int n;
    scanf("%d", &n);

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

    printf("%d\n", longest_non_decreasing_subsequence(sequence, n));
    return 0;
}
```

Output:



	Input	Expected	Got	
✓	9 -1 3 4 5 2 2 2 2 3	6	6	✓
✓	7 1 2 2 4 5 7 6	6	6	✓