Date: 17/09/2020   Deadline:19/09/2020 (7 pm)

a. You have an array A on total N integers: x_1, x_2, x_3, ..., x_N. You have to develop a dynamic programming to find the longest-alternating-increasing-subsequence and its length in the given array. A sequence (y_1, y_2, y_3,...y_k, where k >=1) is called alternating-increasing when following two properties are there
      1. $|y_1|<|y_2|<|y_3|<.....<|y_k|$
      2. The adjacent elements should have alternate signs. i.e., if $y_1 > 0$ then $y_2 < 0$, $y_3 > 0$ and so on. Similarly, if $y_1 < 0$, then $y_2 > 0$ $y_3 < 0$ and so on.
    In array A, there would be no element equal to 0.

    Constraints:
    1<=N<=500 $|x_i|$<=50, x_i not equal to 0.

    Input:
    The first line contains a single integer N, denoting the size of the array. The next line contains N integers, denoting N elements of array A.

    Output:
    Print the longest alternating sub-sequence and its length.

    Example:
    Input:
    1 3 -2 -1 6 -6 -7 11

    Output:
    [1, -2, 6, -7, 11]
    5
    Submit 5a.c

b. You have an array A on total N positive integers: x_1, x_2, x_3, ..., x_N. We are supposed to find longest decreasing-increasing subsequence ((y_1, y_2,..., y_m,...y_k, where k >=1)) from it which follows following criteria.
      1. y_1, y_2,..., y_m is a strictly decreasing sequence (y_1, y_2,..., y_{m-1} may not be present in this sequence)
      2. y_m,...y_k is a strictly increasing sequence (y_{m+1},...y_k may not be present in this sequence)

    Constraints:
    1<=N<=500 $|x_i|$<=1000

Input:
The first line contains a single integer N, denoting the size of the array. The next line contains N positive integers, denoting N elements of array A.

Output:
Print the longest decreasing-increasing sub-sequence and its length.

Example:
Input:
1 7 3 4 6 8 5 9 10
Output:
7 3 4 6 8 9 10 => element 3 is y_m here.
Length = 7
Alternative output
1 3 4 6 8 9 10 => element 1 is y_m here. (y_1, y_2,…, y_{m-1} are not there in this sequence)
Length = 7

Submit 5b.c