14th Jan 2023

First :-

Next Greater Element : : Medium

Given an array **arr[ ]** of size **N** having elements, the task is to find the next greater element for each element of the array in order of their appearance in the array.

Next greater element of an element in the array is the nearest element on the right which is greater than the current element.

If there does not exist next greater of current element, then next greater element for current element is -1. For example, next greater of the last element is always -1.

Example 1:

```
Input:
N = 4, arr[] = [1 3 2 4]
Output:
3 4 4 -1
Explanation:
In the array, the next larger element
to 1 is 3 , 3 is 4 , 2 is 4 and for 4 ?
since it doesn't exist, it is -1.
```

Example 2:

```
Input:
N = 5, arr[] [6 8 0 1 3]
Output:
8 -1 1 3 -1
Explanation:
In the array, the next larger element to
6 is 8, for 8 there is no larger elements
```

```
hence it is -1, for 0 it is 1 , for 1 it
is 3 and then for 3 there is no larger
element on right and hence -1.
```

Your Task:

This is a **function** problem. You only need to complete the function **nextLargerElement()** that takes list of integers **arr[ ]** and **N** as input parameters and returns list of integers of length N denoting the next greater elements for all the corresponding elements in the input array.

Expected Time Complexity : O(N)
Expected Auxiliary Space : O(N)

Constraints:
$1 \le N \le 10^6$
$1 \le A_i \le 10^{18}$

CODE SECTION:-

```
vector<long long> nextLargerElement(vector<long long> arr, int n)
    {
        // Your code here
        vector<long long> ans;
        stack<long long> st;
        for (int i = n - 1; i >= 0; i--)
        {
            long long x = arr[i];
            if (st.empty())
            {
                ans.push_back(-1);
                st.push(x);
            }
            else if (st.top() > x)
            {
                ans.push_back(st.top());
                st.push(x);
            }
            else
            {
                while (st.top() <= x)
                {
                    st.pop();
                    if (st.empty())
                    {
                        ans.push_back(-1);
                        st.push(x);
                        break;
                    }
                }
                if (st.top() > x)
                {
                    ans.push_back(st.top());
                    st.push(x);
                }
            }
        }
        reverse(ans.begin(), ans.end());
        return ans;
    }
```

-: DONE FOR THE DAY :-