17TH JAN 2023

FIRST :-

Next Greater Element 2 : : Medium

Given a circular interger array **arr** of size **N** (i.e ., the next element of **arr [N-1]** is **arr[0]**), return the **next greater number** for every element in **arr**.

The **next greater element** of a number **x** is the **first greater number** to its traversing-order next in the array, which means you could search circularly to find its next greater number. If it doesn't exist, return **-1** for this number.

Example 1:

Input:
N = 3
arr[ ] = {1, 2, 1}
Output: {2, -1, 2}
Explanation: The first 1's next greater number is 2:
The number 2 can't find next greater number.
The second 1's next greater number needs to search circularly, which is also 2.

Example 2:

Input:
N = 5
arr[ ] = {5, 4, 3, 2, 1}
Output: {-1, 5, 5, 5, 5}

Your Task:

You don't need to read input or print anything. Your task is to complete the function **nextGreaterElement()** which takes the array of integers **arr** and **N** as parameters and returns an array of integer which contains the next greater number for every element in arr.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(N)

Constraints:

$1 \le N \le 10^4$

$10^{-9} \le arr_i \le 10^{-9}$

CODE SECTION:-

```
class Solution {
  public:
    vector<int> nextGreaterElement(int n, vector<int>& arr) {
        // code here
       stack<int> stk;
       vector<int> nge(n,-1);
       for(int i = 0;i< n * 2;i++){
           int curr = arr[i % n] ;
           while(!stk.empty() && arr[stk.top()] < curr){
              nge[stk.top()] = curr;
              stk.pop();
           }
           if(i<n)stk.push(i);
       }
       return nge;
    }
};
```

-: Done for the today :-