Next Greater Element

Approach :

We traverse the array backwards push elements into stack one by one comparing whether the element at top is greater than the current element if it is then we store this in the ans for our current element otherwise we keep popping from stack until this condition is met.

If stack becomes empty at any point then the ans array will contain -1 for such elements.

Code :

```
#include <bits/stdc++.h>

vector<int> nextGreater(vector<int> &arr, int n) {
    // Write your code here
    stack<int> s;
    vector<int> ans;
    for(int i=n-1;i>=0;i--)
    {
        while(!s.empty() && s.top()<=arr[i])
        {
            s.pop();
        }
        if(s.empty())
        {
            ans.push_back(-1);
        }
        else
        {
            ans.push_back(s.top());
        }
        s.push(arr[i]);


    }
    reverse(ans.begin(),ans.end());
    return ans;
}
```

- Time Complexity : O(n)

- Space Complexity : O(N)