Next Greater/Smaller Element

Source: GeeksForGeeks, IDeserve

Given an array, print the Next Greater Element (NGE) for every element. The Next greater Element for an element x is the **first** greater element **on the right side** of x in array.

Time Complexity: $O(n)$

```
void nextSmallerElement(vector<int> ar) {
    vector<int> ans(ar.size(), -1);
    stack <pair<ll, int>> s;
    s.push(make_pair(ar[0], 0));
    int len = ar.size();

    for (int i = 1; i < len; ++i) {
        while (!s.empty() && s.top().first > ar[i]) {
            ans[s.top().second] = i;
            s.pop();
        }
        s.push({ar[i], i});
    }
}

void nextGreaterElement(vector<int> ar) {
    vector<int> ans(ar.size(), -1);
    stack <pair<ll, int>> s;
    s.push(make_pair(ar[0], 0));
    int len = ar.size();

    for (int i = 1; i < len; ++i) {
        while (!s.empty() && s.top().first < ar[i]) {
            ans[s.top().second] = i;
            s.pop();
        }
        s.push({ar[i], i});
    }
}
```

```
idx:    0   1   2   3   4   5   6
input:  98  23  54  12  19  19  11
output: 1   3   3   6   6   6   -1
Next Smaller element for 23 is 12 with idx '6'
```

Now Similarly we can do the opposite and find out the first previous greater/smaller element.

```
void prevGreaterElement(vector<int> ar) {
    stack <pair<ll, int>> s;
    int len = ar.size();
    vector<int> ans(len, -1);
    s.push(make_pair(ar[len-1], len-1));

    for (int i = len-1; i >= 0; --i) {
        while (!s.empty() && s.top().first < ar[i]) {
            ans[s.top().second] = i;
            s.pop();
        }
        s.push({ar[i], i});
    }
}
void prevSmallerElement(vector<int> ar) {
    stack <pair<ll, int>> s;
    int len = ar.size();
    vector<int> ans(len, -1);
    s.push(make_pair(ar[len-1], len-1));
    for (int i = len-1; i >= 0; --i) {
        while (!s.empty() && s.top().first > ar[i]) {
            ans[s.top().second] = i;
            s.pop();
        }
        s.push({ar[i], i});
    }

    for (int i = 0; i < len; ++i) cout << i << "\t";
    cout << endl;
    for (auto i : ar) cout << i << "\t";
    puts("");
    for (auto i : ans) cout << i << "\t";
    puts("");
}
```