#15 Days of Coding Challenge
#Day 7

Sliding window maximum
#Leetcode 239

Solution:

```
class Solution {
public:
    vector<int> maxSlidingWindow(vector<int>& nums, int k) {
        int n=nums.size();
        vector<int> ans;
        deque<int> dq;
        for(int i=0;i<n;i++){
            if(!dq.empty() && dq.front()<=i-k) dq.pop_front();
            while(!dq.empty() && nums[dq.back()]<=nums[i]) dq.pop_back();
            dq.push_back(i);
            if(i>=k-1) ans.push_back(nums[dq.front()]);
        }
        return ans;
    }
};
```

Time Complexity: O(n)
Space Complexity: O(k)