# Sliding Window Problem

## 1:Count Subarrays With Score Less Than K:

```
class Solution {
public:
    long long countSubarrays(vector<int>& nums, long long k) {
        long long ans = 0;   // Initialize the answer variable to count the number of subarrays
        long long l = 0;     // Pointer for the left end of the current subarray
        long long curSum = 0;   // Variable to store the current sum of the subarray

        // Iterate through the array
        for(int i = 0; i < nums.size(); i++) {
            curSum += (long long)nums[i];   // Add the current element to the current sum

            // While the current sum multiplied by the length of the subarray is greater than or equal to k
            while(curSum * (i - l + 1) >= k) {
                curSum -= nums[l];   // Remove the leftmost element from the subarray
                l++;   // Move the left pointer to the right
            }

             ans += (long long)(i - l + 1);   // Add the number of valid subarrays ending at index i
         }

         return ans;   // Return the total number of valid subarrays
    }
};

TC :- O(n), SC :- O(1)
```

## 2: Number of subarrays having sum less than K: