1. Remove Duplicates from Sorted Array

You are given a sorted array nums. Remove all duplicates **in-place** (i.e., without using another array) so that each element appears only once.
Return the new length of the array after removing duplicates.

Example:

Input: nums = [1, 1, 2, 2, 3, 3, 3, 4]
Output: [1, 2, 3, 4] (new length = 4)

2. Longest Consecutive Sequence:

Given an unsorted array of integers, find the **length of the longest sequence of consecutive numbers** (numbers that appear one after another in order, not necessarily adjacent in the array).

Example:

Input: nums = [100, 4, 200, 1, 3, 2]
Output: 4
Explanation: The consecutive sequence is [1, 2, 3, 4].

3. 3Sum (Find Triplets that Sum to Zero)

Find all unique triplets (a, b, c) in the array such that a + b + c = 0.

Example:

Input: nums = [-1, 0, 1, 2, -1, -4]
Output: [[-1, 0, 1], [-1, -1, 2]]

4. Move Zeroes

Move all zeros to the end of the array while keeping the order of non-zero elements **the same**.

Example:

Input: nums = [0, 1, 0, 3, 12]
Output: [1, 3, 12, 0, 0]

5. Majority Element

Find the element that appears **more than n/2 times** in the array (it always exists).

Example:

Input: `nums = [2, 2, 1, 1, 1, 2, 2]`
Output: 2

Hint: You can use the **Boyer–Moore Voting Algorithm**. Keep a candidate and a counter. Increase counter when the same element appears, decrease otherwise.

6. Subarray Sum Equals K

Given an array of integers nums and an integer k, count the total number of **continuous subarrays** whose sum equals k.

Example:

Input: `nums = [1, 1, 1], k = 2`
Output: 2
Explanation: Subarrays `[1,1]` (indices 0-1) and `[1,1]` (indices 1-2) both sum to 2.

16. Longest Subarray with Sum ≤ K

Find the **length** of the longest subarray whose sum is **less than or equal to K**.

Example:

Input: `nums = [2, -1, 2, 3, -4, 2], k = 3`
Output: `5`
E**xplanation:** The longest subarray `[2, -1, 2, 3, -4]` has sum `2`.