Assignment-08

1. Reverse a String

Write a JavaScript program that reverses a given string.

Input: "world", Output: "dlrow"

2.Check if a String is a Palindrome

Write a JavaScript program that checks if a given string is a palindrome (reads the same forwards and backwards).

Input: "racecar" Output: true

Input**:** "hello" Output**:** false

3.Check if a Number is Prime

Write a JavaScript program to check if a given number is prime (a prime number is only divisible by 1 and itself).

Input: 11 Output: true

Input: 10 Output: false

4.Find the Second Largest Number in an Array

Write a JavaScript program to find the second largest number in a given array.

Input: [12, 45, 22, 68, 39] Output: 45

5. Remove Duplicates in-place from Sorted Array.

Given a sorted array, remove the duplicates in-place such that each element appears only once, and return the new length of the array.

Example 1:

Input: [1, 1, 2, 2, 2, 3, 4, 4, 5, 5, 5]

Output: 5, [1, 2,3,4,5]

6. Move all Zeros to the end of the array.

Given an array, move all zeros to the end of the array while maintaining the relative order of the non-zero elements. You must do this in-place without using extra space for another array.

Example 1:
Input: `[0, 1, 0, 3, 0, 12]`
Output: `[1, 3, 12, 0, 0, 0]`

7. Find the first occurrence of an element.

Given a sorted array and a target value, return the index of the first occurrence of the target. If the target is not found, return `-1`. Solve this with O(log n) time complexity.

Example 1:
Input: `arr = [1, 2, 2, 2, 3, 4], target = 2`
Output: `1`

8. Count of an element in a sorted array.

Given a sorted array and a target value, return the count of occurrences of the target in the array. If the target is not found, return `0`. Solve this with O(log n) time complexity.

Example 2:
Input: `arr = [5, 7, 7, 8, 8, 10], target = 8`
Output: `2`