### **1. Remove Duplicates from an Array**

**Description:**

Write a function that takes an array and returns a new array with duplicate values removed.

**Example:**

* **Input:** [1, 2, 3, 2, 4, 1]
* **Output:** [1, 2, 3, 4]

### **2. Flatten a Nested Array Recursively**

**Description:**

Create a function that recursively flattens an array with any level of nesting.

**Example:**

* **Input:** [1, [2, [3, 4], 5], 6]
* **Output:** [1, 2, 3, 4, 5, 6]

### **3. Reverse a String Without Using Built-in Reverse Methods**

**Description:**

Reverse a string character by character without using the built-in reverse() method.

**Example:**

* **Input:** "hello"
* **Output:** "olleh"

### **4. Find the Intersection of Two Arrays**

**Description:**

Write a function that returns a new array containing only the common elements between two arrays.

**Example:**

* **Input:** [1, 2, 3, 4] and [3, 4, 5, 6]
* **Output:** [3, 4]

### **5. Merge Two Sorted Arrays**

**Description:**

Merge two sorted arrays into one sorted array (without using the built-in sort() method).

**Example:**

* **Input:** [1, 3, 5] and [2, 4, 6]
* **Output:** [1, 2, 3, 4, 5, 6]

### **6. Find the Union of Two Arrays**

**Description:**

Write a function that returns an array containing the union of two arrays (all unique elements from both arrays).

**Example:**

* **Input:** [1, 2, 3] and [2, 3, 4, 5]
* **Output:** [1, 2, 3, 4, 5]

### **7. Implement Binary Search on a Sorted Array**

**Description:**

Create a function that performs binary search on a sorted array and returns the index of a given target element (or -1 if not found).

**Example:**

* **Input:** Array: [1, 3, 5, 7, 9], Target: 7
* **Output:** 3

### **8. Check if an Array is a Subset of Another**

**Description:**

Write a function that checks if one array is a subset of another array (all elements in the first array exist in the second).

**Example:**

* **Input:** Subset: [2, 3], Superset: [1, 2, 3, 4]
* **Output:** true

### **9. Check if Two Strings are Anagrams**

**Description:**

Write a function to determine if two strings are anagrams (i.e., they contain the same characters in a different order).

**Example:**

* **Input:** "listen", "silent"
* **Output:** true

### **10. Find the Second Largest Number in an Array**

**Description:**

Create a function that returns the second largest number in an unsorted array.

**Example:**

* **Input:** [10, 5, 8, 20, 15]
* **Output:** 15

### **11. Find All Permutations of a String**

**Description:**

Write a function that returns an array of all possible permutations of a given string.

**Example:**

* **Input:** "abc"
* **Output:** ["abc", "acb", "bac", "bca", "cab", "cba"]

### **12. Convert a Number to a Roman Numeral**

**Description:**

Create a function that converts an integer to its corresponding Roman numeral representation.

**Example:**

* **Input:** 1994
* **Output:** "MCMXCIV"

### **13. Find the Longest Common Prefix Among an Array of Strings**

**Description:**

Write a function that takes an array of strings and returns the longest common prefix shared by all strings. If there is no common prefix, return an empty string.

**Example:**

* **Input:** ["flower", "flow", "flight"]
* **Output:** "fl"
* **Input:** ["dog", "racecar", "car"]
* **Output:** ""

### **14. Count Vowels in a String**

**Description:**

Create a function that counts the number of vowels (a, e, i, o, u) in a given string.

**Example:**

* **Input:** "hello world"
* **Output:** 3

### **15. Check if a Number is a Power of Two**

**Description:**

Implement a function that determines if a given number is a power of two.

**Example:**

* **Input:** 16
* **Output:** true
* **Input:** 18
* **Output:** false

### **16. Longest Substring Without Repeating Characters**

**Description:**

Write a function to find the longest substring of a given string that contains no repeating characters. You may return the substring itself or its length.

**Example:**

* **Input:** "abcabcbb"
* **Output:** "abc" (or length 3)

### **17. Flatten a Nested Object to Dot Notation**

**Description:**

Write a function that converts a nested object into a flat object using dot notation for the keys.

**Example:**

* **Input:** { a: { b: { c: 1 } } }
* **Output:** { "a.b.c": 1 }

### **18. Generate a Fibonacci Sequence up to n Elements**

**Description:**

Write a function that returns an array containing the Fibonacci sequence with n elements.

**Example:**

* **Input:** 7
* **Output:** [0, 1, 1, 2, 3, 5, 8]

### **19. Determine if a Year is a Leap Year**

**Description:**

Write a function that checks if a given year is a leap year.

**Rules to Remember:**

* A year is a leap year if it is divisible by 4
* Except if the year is divisible by 100, then it is not a leap year
* However, if the year is divisible by 400, it is a leap year

**Example:**

* **Input:** 2000
* **Output:** true
* **Input:** 1900
* **Output:** false

### **20. Calculate the Factorial of a Number**

**Description:**

Implement a function that calculates the factorial of a given non-negative integer. The factorial of a number n (written as n!) is the product of all positive integers less than or equal to n.

**Example:**

* **Input:** 5
* **Output:** 120 (because 5! = 5 × 4 × 3 × 2 × 1)