✅ Basic Programming (1–20)

1. **FizzBuzz**
   - **Description**: Write a program that prints numbers from 1 to 100. For multiples of three, print "Fizz" instead of the number, and for the multiples of five, print "Buzz". For numbers which are multiples of both three and five, print "FizzBuzz".
   - **Expected Output**:
   - 1
   - 2
   - Fizz
   - 4
   - Buzz
   - Fizz
   - 7
   - 8
   - Fizz
   - Buzz
2. **Two Sum**
   - **Description**: Given an array of integers and a target value, return the indices of the two numbers that add up to the target.
   - **Expected Output**: Input: nums = [2, 7, 11, 15], target = 9 Output: [0, 1]
3. **Reverse Integer**
   - **Description**: Given a 32-bit signed integer, reverse the digits of the integer.
   - **Expected Output**: Input: 123 Output: 321
4. **Palindrome Number**
   - **Description**: Determine if a given integer is a palindrome. An integer is a palindrome when it reads the same backward as forward.
   - **Expected Output**: Input: 121 Output: true
5. **Roman to Integer**
   - **Description**: Given a Roman numeral, convert it to an integer.
   - **Expected Output**: Input: III Output: 3
6. **Integer to Roman**
   - **Description**: Convert an integer to a Roman numeral.
   - **Expected Output**: Input: 58 Output: LVIII
7. **Count and Say**
   - **Description**: The count-and-say sequence is a sequence of numbers where each number is the description of the previous number. Starting with "1", the next number is "11" (one 1), the next one is "21" (two 1s), the next one is "1211" (one 2, one 1), and so on.
   - **Expected Output**: Input: 4 Output: 1211
8. **Power of Two**
   - **Description**: Given an integer, write a function to check if it is a power of two.
   - **Expected Output**: Input: 16 Output: true
9. **Factorial Trailing Zeroes**
   - **Description**: Given an integer n, return the number of trailing zeroes in n!.
   - **Expected Output**: Input: 25 Output: 6
10. **Add Digits (Digital Root)**
    - **Description**: Given a non-negative integer num, repeatedly add all its digits until the result has only one digit.
    - **Expected Output**: Input: 38 Output: 2
11. **Valid Parentheses**
    - **Description**: Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid. An input string is valid if the brackets are closed in the correct order.
    - **Expected Output**: Input: "(){}[]" Output: true

12. Merge Two Sorted Arrays
- **Description**: Given two sorted integer arrays, merge them into one sorted array.
- **Expected Output**: Input: nums1 = [1, 3, 5], nums2 = [2, 4, 6] Output: [1, 2, 3, 4, 5, 6]

13. Remove Duplicates from Sorted Array
- **Description**: Given a sorted array, remove the duplicates in place such that each element appears only once and return the new length.
- **Expected Output**: Input: [1, 1, 2] Output: 2

14. Remove Element
- **Description**: Given an array and a value, remove all instances of that value in-place and return the new length.
- **Expected Output**: Input: nums = [3, 2, 2, 3], val = 3 Output: 2

15. Find the Index of First Occurrence
- **Description**: Given a string and a target character, return the index of the first occurrence of the character in the string.
- **Expected Output**: Input: "hello", target = 'e' Output: 1

16. Length of Last Word
- **Description**: Given a string consisting of words separated by spaces, find the length of the last word.
- **Expected Output**: Input: "Hello World" Output: 5

17. Climbing Stairs
- **Description**: You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. How many distinct ways can you climb to the top?
- **Expected Output**: Input: 3 Output: 3

18. Maximum Subarray
- **Description**: Given an integer array, find the contiguous subarray (containing at least one number) that has the largest sum and return its sum.
- **Expected Output**: Input: [-2, 1, -3, 4, -1, 2, 1, -5, 4] Output: 6

19. Plus One
- **Description**: Given a non-empty array of digits representing a non-negative integer, add one to the integer and return the resulting array.
- **Expected Output**: Input: [1, 2, 3] Output: [1, 2, 4]

20. Search Insert Position
- **Description**: Given a sorted array of distinct integers and a target value, return the index where the target should be inserted.
- **Expected Output**: Input: [1, 3, 5, 6], target = 5 Output: 2

String Problems (21–40)

---

21. Longest Common Prefix
- **Description**: Write a function to find the longest common prefix string amongst an array of strings. If there is no common prefix, return an empty string.
- **Expected Output**: Input: ["flower", "flow", "flight"] Output: "fl"

22. Valid Anagram
- **Description**: Given two strings, write a function to determine if one string is an anagram of the other.
- **Expected Output**: Input: s = "anagram", t = "nagaram" Output: true

23. Implement strStr()
- **Description**: Implement the strStr() function. It should return the index of the first occurrence of the substring needle in the string haystack. Return -1 if the needle is not part of the haystack.
- **Expected Output**: Input: haystack = "hello", needle = "ll" Output: 2

24. Isomorphic Strings
- **Description**: Given two strings, determine if they are isomorphic. Two strings are isomorphic if the characters in one string can be replaced to get the other string.
- **Expected Output**: Input: "egg", "add" Output: true

25. Reverse String
- **Description**: Write a function that reverses a string.
- **Expected Output**: Input: "hello" Output: "olleh"

26. Reverse Words in a String
- **Description**: Given an input string, reverse the order of the words.
- **Expected Output**: Input: "the sky is blue" Output: "blue is sky the"

27. Check for Palindrome String
- **Description**: Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.
- **Expected Output**: Input: "A man, a plan, a canal: Panama" Output: true

28. Group Anagrams
- **Description**: Given an array of strings, group the anagrams together.
- **Expected Output**: Input: ["eat", "tea", "tan", "ate", "nat", "bat"] Output: [["eat", "tea", "ate"], ["tan", "nat"], ["bat"]]

29. Longest Palindromic Substring
- **Description**: Given a string, find the longest substring that is a palindrome.
- **Expected Output**: Input: "babad" Output: "bab"

30. First Unique Character in a String
- **Description**: Given a string, find the first non-repeating character and return its index.
- **Expected Output**: Input: "leetcode" Output: 0

31. Count Binary Substrings
- **Description**: Given a string consisting of characters '0' and '1', count the number of substrings with equal number of '0's and '1's.
- **Expected Output**: Input: "00110011" Output: 6

32. String Compression
- **Description**: Given a string, compress it using the counts of repeated characters.
- **Expected Output**: Input: "aaabbbccc" Output: "a3b3c3"

33. Check if Two Strings are One Edit Distance Apart
- **Description**: Given two strings, check if they are one edit distance apart. An edit is defined as inserting a character, deleting a character, or replacing a character.
- **Expected Output**: Input: "abc", "ab" Output: true

34. Check if a String is a Subsequence
- **Description**: Given two strings, check if the second string is a subsequence of the first string.
- **Expected Output**: Input: "abc", "ac" Output: true

35. Check if a Sentence is a Pangram
- **Description**: Given a sentence, check if it contains every letter of the alphabet at least once.
- **Expected Output**: Input: "the quick brown fox jumps over the lazy dog" Output: true

36. Count the Occurrences of a Word
- **Description**: Given a string, count the number of occurrences of a given word.
- **Expected Output**: Input: "hello world, hello" Output: 2

37. Detect Capital Usage
- **Description**: Given a string, determine if the usage of capitals is correct. The usage is correct if all letters are capitals, all are lowercase, or if the first letter is uppercase and the rest are lowercase.
- **Expected Output**: Input: "USA" Output: true

38. Implement atoi()
- **Description**: Implement the atoi() function, which converts a string to an integer.
- **Expected Output**: Input: "42" Output: 42

39. Decode String
- **Description**: Given an encoded string, decode it. The encoded string is a string containing digits followed by a string, and you must repeat the string as many times as the number of digits.
- **Expected Output**: Input: "3[a]2[bc]" Output: "aaabcbc"

40. Find All Duplicates in a String

- **Description**: Given a string, find all the characters that appear more than once in the string.
- **Expected Output**: Input: "aabcc" Output: ["a", "c"]

🔁 Array Problems (41–60)

---

41. Move Zeroes

- **Description**: Given an array of integers, move all zeroes to the end without changing the order of non-zero elements.
- **Expected Output**: Input: [0, 1, 0, 3, 12] Output: [1, 3, 12, 0, 0]

42. Rotate Array

- **Description**: Given an array, rotate the elements of the array to the right by k steps.
- **Expected Output**: Input: [1, 2, 3, 4, 5, 6, 7], k = 3 Output: [5, 6, 7, 1, 2, 3, 4]

43. Intersection of Two Arrays

- **Description**: Given two arrays, return their intersection (the elements that appear in both arrays).
- **Expected Output**: Input: [1, 2, 2, 1], [2, 2] Output: [2, 2]

44. Two Sum II (Sorted Array)

- **Description**: Given a sorted array and a target value, return the indices of two numbers such that their sum is equal to the target.
- **Expected Output**: Input: numbers = [2, 7, 11, 15], target = 9 Output: [1, 2]

45. Maximum Product of Two Elements

- **Description**: Given an array, find the maximum product of two elements.
- **Expected Output**: Input: [1, 2, 3, 4] Output: 12

46. Single Number

- **Description**: Given an array of integers, every element appears twice except for one. Find that single element.
- **Expected Output**: Input: [2, 2, 1] Output: 1

47. Find Missing Number

- **Description**: Given an array containing n distinct numbers taken from the range 1 to n + 1, find the one number that is missing from the array.
- **Expected Output**: Input: [3, 7, 1, 2, 8, 4, 5] Output: 6

48. Contains Duplicate

- **Description**: Given an array, check if any value appears at least twice.
- **Expected Output**: Input: [1, 2, 3, 1] Output: true

49. Third Maximum Number

- **Description**: Given an array of integers, return the third maximum number. If it does not exist, return the maximum number.
- **Expected Output**: Input: [3, 2, 1] Output: 1

50. Maximum Consecutive Ones

- **Description**: Given a binary array, find the maximum number of consecutive 1's.
- **Expected Output**: Input: [1, 1, 0, 1, 1, 1] Output: 3

51. Merge Intervals

- **Description**: Given a collection of intervals, merge all overlapping intervals.
- **Expected Output**: Input: [[1, 3], [2, 6], [8, 10], [15, 18]] Output: [[1, 6], [8, 10], [15, 18]]

52. Best Time to Buy and Sell Stock

- **Description**: Given an array of prices, where prices[i] is the price of a given stock on the i-th day, find the maximum profit you can achieve from a single buy and sell operation.
- **Expected Output**: Input: [7, 1, 5, 3, 6, 4] Output: 5

53. Majority Element

- **Description**: Given an array of size n, find the majority element. The majority element is the element that appears more than n / 2 times.
- **Expected Output**: Input: [3, 2, 3] Output: 3

54. Find All Numbers Disappeared in Array

- **Description**: Given an array of integers, some numbers appear twice, and others appear once. Find all the numbers that are missing from the array.
- **Expected Output**: Input: [4, 3, 2, 7, 8, 2, 3, 1] Output: [5, 6]

55. Set Matrix Zeroes

- **Description**: Given an m x n matrix, if an element is 0, set its entire row and column to 0.
- **Expected Output**: Input: [[1, 1, 1], [1, 0, 1], [1, 1, 1]] Output: [[1, 0, 1], [0, 0, 0], [1, 0, 1]]

56. Spiral Matrix

- **Description**: Given an m x n matrix, return all the elements of the matrix in spiral order.
- **Expected Output**: Input: [[1, 2, 3], [4, 5, 6], [7, 8, 9]] Output: [1, 2, 3, 6, 9, 8, 7, 4, 5]

57. Pascal's Triangle

- **Description**: Given an integer numRows, return the first numRows of Pascal's triangle.
- **Expected Output**: Input: 5 Output: [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]

58. Find Pivot Index

- **Description**: Given an array of integers, find the pivot index, which is the index where the sum of the numbers on the left is equal to the sum of the numbers on the right.
- **Expected Output**: Input: [1, 7, 3, 6, 5, 6] Output: 3

59. Product of Array Except Self

- **Description**: Given an array of integers, return an array where each element is the product of all the elements of the input array except the element at that position.
- **Expected Output**: Input: [1, 2, 3, 4] Output: [24, 12, 8, 6]

60. Valid Mountain Array

- **Description**: Given an array, determine if it is a mountain array. A mountain array is an array where the values strictly increase and then strictly decrease.
- **Expected Output**: Input: [2, 1] Output: false

📐 **Math & Number Theory (61–70)**

---

61. Sqrt(x) without using built-in function

- **Description**: Given a non-negative integer x, implement the sqrt(x) function, which returns the square root of x (rounded down).
- **Expected Output**: Input: 8 Output: 2

62. Check if Number is Power of Three

- **Description**: Given an integer, check if it is a power of three.
- **Expected Output**: Input: 27 Output: true

63. Find GCD of Two Numbers

- **Description**: Given two integers, find the greatest common divisor (GCD) of the two numbers.
- **Expected Output**: Input: 8, 12 Output: 4

64. Count Primes Less Than N

- **Description**: Given an integer n, return the number of prime numbers that are strictly less than n.
- **Expected Output**: Input: 10 Output: 4

65. Ugly Number

- **Description**: An ugly number is a number whose prime factors only include 2, 3, or 5. Given an integer n, determine if it is an ugly number.
- **Expected Output**: Input: 6 Output: true

66. Happy Number

- **Description**: A happy number is defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 or it loops endlessly in a cycle.
- **Expected Output**: Input: 19 Output: true

67. Sum of Two Integers Without + or -

- **Description**: Given two integers a and b, return their sum without using the operators + and -.
- **Expected Output**: Input: 1, 2 Output: 3

68. Excel Sheet Column Title

- **Description**: Given a positive integer, convert it to its corresponding column title as it appears in an Excel sheet.
- **Expected Output**: Input: 28 Output: "AB"

69. Reverse Bits

- **Description**: Given a 32-bit unsigned integer, reverse the bits of the number and return the result.
- **Expected Output**: Input: 43261596 Output: 964176192

70. Hamming Distance

- **Description**: The Hamming distance between two integers is the number of positions at which the corresponding bits are different. Given two integers, find their Hamming distance.
- **Expected Output**: Input: 1, 4 Output: 2

🎯 **Search and Sorting (71–80)**

---

71. Binary Search

- **Description**: Implement binary search. Given a sorted array of integers and a target value, return the index of the target value if it exists, otherwise return -1.
- **Expected Output**: Input: [-1, 0, 3, 5, 9, 12], target = 9 Output: 4

72. Linear Search

- **Description**: Implement linear search. Given an array and a target value, return the index of the target value if it exists, otherwise return -1.
- **Expected Output**: Input: [2, 3, 4, 5, 6], target = 4 Output: 2

73. Insertion Sort

- **Description**: Implement the insertion sort algorithm to sort an array of integers in ascending order.
- **Expected Output**: Input: [5, 2, 4, 6, 1, 3] Output: [1, 2, 3, 4, 5, 6]

74. Selection Sort

- **Description**: Implement the selection sort algorithm to sort an array of integers in ascending order.
- **Expected Output**: Input: [5, 2, 4, 6, 1, 3] Output: [1, 2, 3, 4, 5, 6]

75. Merge Sort

- **Description**: Implement the merge sort algorithm to sort an array of integers in ascending order.
- **Expected Output**: Input: [5, 2, 4, 6, 1, 3] Output: [1, 2, 3, 4, 5, 6]

76. Quick Sort

- **Description**: Implement the quick sort algorithm to sort an array of integers in ascending order.
- **Expected Output**: Input: [5, 2, 4, 6, 1, 3] Output: [1, 2, 3, 4, 5, 6]

77. Bubble Sort

- **Description**: Implement the bubble sort algorithm to sort an array of integers in ascending order.
- **Expected Output**: Input: [5, 2, 4, 6, 1, 3] Output: [1, 2, 3, 4, 5, 6]

78. Find First and Last Position of Target

- **Description**: Given a sorted array of integers and a target value, return the first and last position of the target in the array. If the target does not exist, return [-1, -1].
- **Expected Output**: Input: [5, 7, 7, 8, 8, 10], target = 8 Output: [3, 4]

79. Search in Rotated Sorted Array

- **Description**: Given a rotated sorted array and a target value, search for the target and return its index, or -1 if it does not exist.
- **Expected Output**: Input: [4, 5, 6, 7, 0, 1, 2], target = 0 Output: 4

80. Find Peak Element

- **Description**: Given an array, find a peak element. A peak element is an element that is greater than its neighbors.
- **Expected Output**: Input: [1, 2, 3, 1] Output: 3

🔢 HashMap & Sets (81–90)

81. Two Sum (with Map)
- **Description**: Given an array of integers and a target value, find two numbers that add up to the target value. Return their indices.
- **Expected Output**: Input: [2, 7, 11, 15], target = 9 Output: [0, 1]

82. Subarray Sum Equals K
- **Description**: Given an array of integers and an integer k, find the number of continuous subarrays whose sum equals k.
- **Expected Output**: Input: nums = [1, 1, 1], k = 2 Output: 2

83. Longest Consecutive Sequence
- **Description**: Given an unsorted array of integers, find the length of the longest consecutive elements sequence.
- **Expected Output**: Input: [100, 4, 200, 1, 3, 2] Output: 4

84. First Repeating Element
- **Description**: Given an array, find the first repeating element. If no element is repeating, return -1.
- **Expected Output**: Input: [1, 2, 3, 4, 5, 6, 7, 8, 3] Output: 3

85. Longest Substring Without Repeating Characters
- **Description**: Given a string, find the length of the longest substring without repeating characters.
- **Expected Output**: Input: "abcabcbb" Output: 3

86. Group Shifted Strings
- **Description**: Given a list of strings, group them if they are shifted strings (i.e., they can be made from each other by shifting their characters).
- **Expected Output**: Input: ["abc", "bcd", "acef", "aefz", "xyz", "az", "ba", "a", "z"] Output: [["acef", "aefz"], ["abc", "bcd", "xyz"], ["az", "ba"], ["a"], ["z"]]

87. Intersection of Arrays Using Set
- **Description**: Given two arrays, return their intersection using set operations.
- **Expected Output**: Input: [1, 2, 2, 1], [2, 2] Output: [2]

88. Ransom Note
- **Description**: Given a ransom note string and a magazine string, check if you can construct the ransom note from the magazine. Each letter in the magazine can only be used once in the ransom note.
- **Expected Output**: Input: ransomNote = "a", magazine = "b" Output: false

89. Word Pattern
- **Description**: Given a pattern and a string, check if the string follows the same pattern.
- **Expected Output**: Input: pattern = "abba", str = "dog cat cat dog" Output: true

90. Find Duplicate Subtrees (Advanced Beginner)
- **Description**: Given a binary tree, find all duplicate subtrees (subtrees that appear more than once).
- **Expected Output**: Input: root = [1, 2, 3, 4, null, 2, 4, null, null, 4] Output: [[2, 4], [4]]

🧠 Logic & Miscellaneous (91–100)

91. Implement a Queue using Stacks
- **Description**: Implement a queue using two stacks. Implement methods enqueue and dequeue.
- **Expected Output**: Input: enqueue(1), enqueue(2), dequeue() Output: 1

92. Implement a Stack using Queues
- **Description**: Implement a stack using two queues. Implement methods push and pop.
- **Expected Output**: Input: push(1), push(2), pop() Output: 2

93. Valid Sudoku
- **Description**: Given a partially filled Sudoku board, determine if it is valid.
- **Expected Output**: Input: board = [["5", "3", ".", ".", "7", ".", ".", ".", "."], ["6", ".", ".", "1", "9", "5", ".", ".", "."], ...] Output: true

94. Count Steps to Reduce Number to Zero

- **Description**: Given a non-negative integer num, find the number of steps to reduce it to zero. In one step, you can subtract 1 from num or divide num by 2 or 3 if it is divisible.
- **Expected Output**: Input: num = 8 Output: 4

95. Minimum Value to Get Positive Step by Step Sum

- **Description**: Given an array of integers, find the minimum positive value x that you must add to each element to make the cumulative sum always positive.
- **Expected Output**: Input: nums = [-1, -2, -3] Output: 6

96. Check if a Matrix is Toeplitz

- **Description**: A matrix is called Toeplitz if every diagonal from top-left to bottom-right has the same element. Check if a given matrix is Toeplitz.
- **Expected Output**: Input: matrix = [[1, 2, 3, 4], [5, 1, 2, 3], [9, 5, 1, 2]] Output: true

97. Transpose of a Matrix

- **Description**: Given a matrix, return the transpose of the matrix.
- **Expected Output**: Input: matrix = [[1, 2, 3], [4, 5, 6]] Output: [[1, 4], [2, 5], [3, 6]]

98. Diagonal Sum of a Matrix

- **Description**: Given a square matrix, return the sum of the diagonals.
- **Expected Output**: Input: matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]] Output: 25

99. Matrix Reshape

- **Description**: Given a matrix, reshape it into a new matrix with a different number of rows and columns.
- **Expected Output**: Input: matrix = [[1, 2], [3, 4]], r = 1, c = 4 Output: [[1, 2, 3, 4]]

100. Flatten a 2D Array

- **Description**: Given a 2D array, flatten it into a 1D array.
- **Expected Output**: Input: arr = [[1, 2], [3, 4]] Output: [1, 2, 3, 4]
