Mastering DSA — Practice Tasks (No AI Tools)

1. Arrays & Strings (Warm-up Zone)

- Reverse an array
- Find max & min element
- Find the second largest element
- Check if the array is sorted
- Remove duplicates from a sorted array (in-place)
- Left rotate an array by k positions
- Move all zeros to the end
- Merge two sorted arrays
- Find missing number in 1…n sequence
- Kadane's Algorithm (maximum subarray sum)

Strings

- Check palindrome string
- Count vowels/consonants
- Reverse words in a sentence
- Check if two strings are anagrams
- Longest common prefix
- Remove duplicate characters
- Find first non-repeating character

2. Hashing & Maps

- Count frequency of each element
- Find element appearing once when others appear twice
- Two Sum problem (pair of elements with given sum)
- Longest subarray with zero sum
- Subarray with given sum (positive numbers)
- Intersection and Union of two arrays
- Find first repeating and non-repeating element

3. Recursion & Backtracking

- Factorial of a number (recursive)
- Print Fibonacci sequence recursively
- Sum of digits recursively

- Generate all subsets of an array
- Generate all permutations of a string
- Solve N-Queens problem (start with N=4)
- Rat in a maze / Sudoku solver

4. Linked Lists

- Create and print a linked list
- Reverse a linked list
- Find middle of linked list (tortoise-hare method)
- Detect loop in linked list (Floyd’s cycle detection)
- Merge two sorted linked lists
- Remove Nth node from end
- Check palindrome linked list

5. Stacks & Queues

- Implement stack using array / linked list
- Check balanced parentheses ()[]{}
- Next Greater Element
- Evaluate postfix expression
- Min stack (track min in O(1))
- Implement queue using array / linked list
- Circular queue
- Implement stack using two queues (and vice versa)
- Sliding window maximum

6. Trees & Binary Search Trees

- Inorder / Preorder / Postorder traversal
- Level order traversal (BFS)
- Height of a tree
- Count leaf nodes
- Check if tree is balanced
- Lowest Common Ancestor (LCA)
- Validate BST
- Insert / Delete node in BST

7. Graphs

- Represent a graph (adjacency list)

- BFS and DFS traversals
- Detect cycle in directed/undirected graph
- Count connected components
- Topological sort
- Dijkstra’s shortest path

8. Dynamic Programming

- Fibonacci with memoization
- Climbing stairs (ways to reach Nth step)
- Coin change
- 0/1 Knapsack
- Longest Common Subsequence
- Longest Increasing Subsequence