

150 DSA Practice Questions in C Language

Easy (1-50)

Find the largest element in an array.

Find the second largest element in an array.

Check if an array is sorted.

Reverse an array.

Left rotate an array by one place.

Remove duplicates from a sorted array.

Insert an element in an array.

Implement linear search.

Count the number of vowels in a string.

Check if a string is a palindrome.

Find the factorial of a number using recursion.

Print Fibonacci series up to N.

Find the maximum and minimum in an array.

Implement binary search.

Count digits in a number.

Reverse a number.

Check for a prime number.

Find the GCD of two numbers.

Implement selection sort.

Implement bubble sort.

Find the sum of digits of a number.

Find the missing number in an array from 1 to N.

Merge two sorted arrays.

Move all zeros to the end.

Check for anagram strings.

Remove duplicates from an unsorted array.

Count frequency of each element in an array.

Find the longest word in a string.

Convert lowercase to uppercase and vice versa.

Count words in a string.

Find the majority element in an array.

Find common elements in two arrays.

Reverse words in a string.

Calculate power using recursion.

Find the smallest missing positive integer.

Merge two strings alternatively.

Sum of all elements in a matrix.

Transpose of a matrix.

Search for an element in a matrix.

Diagonal sum of a matrix.

Rotate a matrix by 90 degrees (only printing).

Find duplicates in an array.

Check if all characters in a string are unique.

Find first non-repeating character.

Implement `strrev()` manually.

Convert string to integer.

Print all even numbers in an array.

Print characters with even frequency.

Check if a string is valid binary.

Count spaces and special characters in a string.

Medium (51-110)

Implement binary search recursively.

Find the kth largest element in an array.

Rotate array by K elements.

Check if a string is a rotation of another.

Merge intervals.

Longest subarray with sum = 0.

Kadane's algorithm - max subarray sum.

Subarray with given sum.

Find leader elements in an array.

Implement stack using array.

Implement queue using array.

Implement circular queue.

Convert infix to postfix expression.

Evaluate postfix expression.

Check balanced parentheses using stack.

Reverse a stack using recursion.

Sort a stack using another stack.

Implement a min stack.

Implement priority queue.

Implement singly linked list.

Reverse a linked list.

Detect loop in linked list.

Remove loop in linked list.

Find middle of linked list.

Merge two sorted linked lists.

Add two numbers represented by linked lists.

Detect palindrome in linked list.

Find intersection point of two linked lists.

Check for circular linked list.

Find Nth node from the end.

Clone a linked list with random pointers.

Implement doubly linked list.

Implement deque using linked list.

Find union of two arrays.

Find intersection of two arrays.

Longest consecutive subsequence.

Two sum problem.

Sort 0s, 1s, and 2s in an array.

Count number of subarrays with given XOR.

Next greater element.

Implement binary search tree.

Inorder traversal of BST.

Preorder traversal of BST.

Postorder traversal of BST.

Level order traversal of BST.

Check if tree is BST.

Insert a node in BST.

Delete a node in BST.

Check if two trees are identical.

Convert sorted array to BST.

Print all leaf nodes of binary tree.

Find height of binary tree.

Count nodes in a binary tree.

Find diameter of binary tree.

Check if binary tree is balanced.

Print left view of binary tree.

Print right view of binary tree.

Lowest common ancestor in BST.

Print all root-to-leaf paths.

Check for mirror trees.

Hard (111-150)

Construct binary tree from inorder and preorder.

Construct binary tree from inorder and postorder.

Flatten binary tree to linked list.

Serialize and deserialize binary tree.

Boundary traversal of binary tree.

Vertical order traversal of binary tree.

Top view of binary tree.

Bottom view of binary tree.

Morris inorder traversal.

Implement trie (prefix tree).

Word search in matrix (backtracking).

Rat in a maze (backtracking).

N-Queens problem.

Sudoku solver.

Knight's tour problem.

Maximum path sum in binary tree.

Longest palindromic substring.

Longest increasing subsequence.

Matrix chain multiplication.

Edit distance between two strings.

Word break problem.

Minimum number of jumps to reach end.

Count number of islands.

Detect cycle in directed graph.

Detect cycle in undirected graph.

Topological sort of a DAG.

Shortest path in unweighted graph (BFS).

Dijkstra's algorithm.

Bellman-Ford algorithm.

Prim's algorithm for MST.

Kruskal's algorithm for MST.

Disjoint set union (DSU) with path compression.

Find articulation points in graph.

Bridges in a graph (Tarjan's algorithm).

Kosaraju's algorithm for SCC.

Find strongly connected components.

Maximum flow using Ford-Fulkerson.

Traveling salesman problem.

Implement LRU cache.

Trapping rainwater problem.