LogicWhile

Address: MIG 87, First Floor, KPHB Phase I, Hyderabad – 500072
Mobile No: +91 89857 02244, 85898 02244.

Number Programming Questions

1. Check if a number is even or odd
2. Check if a number is positive, negative, or zero
3. Find the maximum of three numbers
4. Swap two numbers (using temp and without temp)
5. Count the number of digits in an integer
6. Reverse a number
7. Check if a number is a palindrome
8. Check if a number is an Armstrong number
9. Check if a number is a perfect number
10. Check if a number is a Harshad (Niven) number
11. Check if a number is prime
12. Print all prime numbers in a given range
13. Find the nth prime number
14. Count total prime numbers between 1 and n
15. Find the next prime number
16. Prime factorization of a number
17. Sieve of Eratosthenes implementation
18. Sum of prime numbers up to n
19. Largest prime factor of a number
20. Find twin primes in a range
21. Check if a number is a strong number
22. Check if a number is an automorphic number
23. Print Fibonacci series up to n terms
24. Check if a number is in Fibonacci sequence
25. Find the nth Fibonacci number
26. Find GCD (HCF) of two numbers
27. Find LCM of two numbers
28. Convert decimal to binary
29. Convert binary to decimal
30. Convert number to Roman numerals
31. Calculate power without using pow()
32. Calculate factorial (iterative and recursive)
33. Count trailing zeros in factorial
34. Compute nCr (combinations)
35. Compute nPr (permutations)
36. Check if a number is strongly prime
37. Check if a number is a happy number
38. Sum of digits of a number
39. Check if a number is a Kaprekar number
40. Find digital root of a number
41. Count number of 1s in binary representation
42. Check if a number is power of 2

43. Find XOR from 1 to n
44. Swap two numbers using XOR
45. Check if a number is sparse (no two consecutive 1s)
46. Find the unique number in array where all except one are repeated
47. Check if binary representation is palindrome
48. Count set bits in a number
49. Add two binary numbers
50. Multiply two numbers without using * operator

Patterns

1. Solid Rectangle Pattern
2. Hollow Rectangle Pattern
3. Square Pattern with Numbers
4. Half Pyramid using Stars
5. Half Pyramid using Numbers
6. Half Pyramid using Alphabets
7. Inverted Half Pyramid using Stars
8. Inverted Half Pyramid using Numbers
9. Inverted Half Pyramid using Alphabets
10. Right-Aligned Star Pyramid
11. Left-Aligned Star Pyramid
12. Full Pyramid (Centered Stars)
13. Inverted Full Pyramid (Centered Stars)
14. Diamond Pattern (Stars)
15. Hollow Diamond Pattern
16. Butterfly Pattern
17. Hollow Butterfly Pattern
18. Sandglass Star Pattern
19. Hourglass Pattern
20. Zig-Zag Pattern
21. Pascal’s Triangle
22. Floyd's Triangle
23. Binary Triangle (0-1 Pattern)
24. Palindromic Number Pyramid
25. Numeric Pyramid (Centered)
26. Numeric Pyramid with Increasing Numbers
27. Numeric Pyramid with Repeated Numbers
28. Inverted Numeric Pyramid
29. Hollow Number Pyramid
30. Concentric Rectangle Number Pattern
31. Cross or X Pattern (Stars)
32. Plus (+) Pattern
33. Alphabet Pyramid (A to A+N)

34. Inverted Alphabet Pyramid
35. Alphabet Half Pyramid
36. Alphabet Half Pyramid Inverted
37. Pyramid with Repeating Alphabets
38. Hollow Alphabet Pyramid
39. Alphabet Diamond Pattern
40. Hollow Square with Diagonals
41. Number Square with Diagonals
42. Snake Number Pattern
43. Spiral Number Matrix
44. Zigzag Character Pattern
45. Triangular Spiral Pattern
46. Rhombus Pattern (Solid)
47. Hollow Rhombus Pattern
48. Mirrored Rhombus Pattern
49. Star Pattern in Letter ‘E’ Shape
50. Star Pattern in Letter ‘Z’ Shape

Arrays

1. Find the maximum and minimum element in an array
2. Find the kth smallest and largest element
3. Reverse the array
4. Left rotate the array by one
5. Right rotate the array by one
6. Left rotate the array by D positions
7. Move all zeros to the end
8. Move all negative elements to one side
9. Find the Union of two arrays
10. Find the Intersection of two arrays
11. Check if two arrays are equal or not
12. Sort an array of 0s, 1s, and 2s (Dutch National Flag)
13. Find the missing number in the array (1 to N)
14. Find all duplicates in an array
15. Find the first repeating element
16. Find the first non-repeating element
17. Check if an array is a palindrome
18. Count the number of occurrences of a number
19. Find the largest sum contiguous subarray (Kadane’s Algorithm)
20. Find the subarray with a given sum
21. Find all subarrays with 0 sum
22. Find the longest subarray with a given sum
23. Find the length of the longest arithmetic subarray
24. Check if array is sorted and rotated

25. Find the equilibrium index of an array
26. Rearrange array in alternating positive & negative items
27. Rearrange array in max/min form
28. Find the majority element (Moore's Voting)
29. Find the element that appears once when all others appear twice
30. Find the element that appears thrice when others appear once
31. Merge two sorted arrays without extra space
32. Merge overlapping intervals
33. Count inversions in an array
34. Find leaders in an array
35. Maximum difference with larger element after smaller
36. Find the ceiling and floor of a number in sorted array
37. Rotate array using reversal algorithm
38. Product of array except self
39. Longest increasing subsequence
40. Longest contiguous increasing subarray
41. Find the median of two sorted arrays
42. Count the number of pairs with a given sum
43. Triplet sum in array
44. Count triplets with sum smaller than target
45. Find common elements in three sorted arrays
46. Maximum product subarray
47. Stock buy and sell to maximize profit
48. Trapping Rain Water
49. Maximum sum of non-adjacent elements
50. Sliding window maximum

2D Arrays

1. Input and print a 2D array
2. Transpose of a matrix
3. Rotate matrix by 90 degrees clockwise
4. Rotate matrix by 180 degrees
5. Reverse each row of the matrix
6. Reverse each column of the matrix
7. Print matrix in spiral order
8. Search an element in a sorted matrix
9. Add two matrices
10. Subtract two matrices
11. Multiply two matrices
12. Diagonal sum (primary and secondary)
13. Check if matrix is symmetric
14. Check if matrix is an identity matrix
15. Count number of zeros in matrix

16. Print boundary elements of the matrix
17. Find row with maximum 1s
18. Find column with maximum sum
19. Find maximum and minimum element in matrix
20. Sort each row of the matrix
21. Sort each column of the matrix
22. Convert matrix to 1D array
23. Print elements in zigzag (wave) form
24. Check if matrix is a magic square
25. Sum of upper triangle of matrix
26. Sum of lower triangle of matrix
27. Snake pattern in matrix
28. Boolean matrix problem (set row/column to 1 if an element is 1)
29. Set entire row and column to 0 if an element is 0
30. Find common elements in all rows of a matrix

Strings

1. Reverse a string
2. Check if a string is a palindrome
3. Check if two strings are anagrams
4. Count vowels and consonants
5. Remove all vowels from a string
6. Convert string to uppercase/lowercase
7. Toggle case of each character
8. Count frequency of each character
9. Find the first non-repeating character
10. Find the first repeating character
11. Remove all duplicate characters
12. Print all substrings of a string
13. Check if one string is a rotation of another
14. Check if string contains only digits
15. Replace all spaces with hyphens/underscores
16. Count words in a sentence
17. Capitalize first letter of each word
18. Reverse each word in a sentence
19. Remove all white spaces
20. Check if a string is a valid palindrome (ignoring non-alphanumerics)
21. Find the longest word in a sentence
22. Find the shortest word in a sentence
23. Remove punctuation from string
24. Print characters at even and odd indices
25. Find the most frequent character
26. Sort characters in a string

27. Find the longest common prefix
28. Find the longest palindromic substring
29. Count occurrences of a word in a sentence
30. Replace a word in a sentence
31. Find all permutations of a string
32. Implement strstr() or indexOf()
33. Check if string is a subsequence of another
34. Check if two strings are isomorphic
35. Remove consecutive duplicate characters
36. Encode string using run-length encoding
37. Decode a run-length encoded string
38. Check if a string has balanced parentheses
39. Convert string to integer (like atoi)
40. Convert integer to string
41. Check if a string is a valid identifier
42. Find duplicate words in a sentence
43. Count uppercase and lowercase letters
44. Check if two strings are meta strings
45. Count the number of palindromic substrings
46. Count the number of words ending with a specific character
47. Convert Roman numerals to integer
48. Convert integer to Roman numerals
49. Find minimum characters to make a string palindrome
50. Find longest substring without repeating characters

Collections

1. Create and iterate over an ArrayList
2. Sort ArrayList (String, Integer)
3. Sort ArrayList of custom objects using Comparable
4. Sort ArrayList using Comparator (custom order)
5. Remove duplicates from ArrayList using Set
6. Merge two ArrayLists and remove duplicates
7. Search an element in ArrayList
8. Convert ArrayList to Array and vice versa
9. Clone an ArrayList
10. Reverse elements of an ArrayList
11. Use LinkedList as a queue
12. Add/remove elements at start/end of LinkedList
13. Convert LinkedList to ArrayList
14. Use Collections.shuffle(), reverse(), and sort()
15. Find frequency of elements using Collections.frequency()
16. Remove duplicates from list using HashSet
17. Find union of two sets

18. Find intersection of two sets
19. Find difference between two sets
20. Check if two sets are equal
21. Sort a set using TreeSet
22. Maintain insertion order using LinkedHashSet
23. Find duplicate characters in a string using Set
24. Store unique words from a sentence
25. Check if two arrays/lists have common elements using Set
26. Count frequency of characters in a string
27. Count frequency of words in a sentence
28. Sort HashMap by keys
29. Sort HashMap by values
30. Find first non-repeating character using LinkedHashMap
31. Check if two HashMaps are equal
32. Merge two HashMaps
33. Remove entries from HashMap based on condition
34. Iterate over keys, values, entries
35. Create custom object as key in HashMap (with equals() and hashCode())
36. Implement stack using Stack class (push, pop, peek)
37. Reverse a string using Stack
38. Balanced parentheses using Stack
39. Evaluate postfix expression using Stack
40. Queue operations using LinkedList
41. Implement circular queue using ArrayDeque
42. Find first non-repeating character in stream using Queue
43. Use PriorityQueue (min-heap)
44. Use PriorityQueue as max-heap (custom Comparator)
45. Find k largest/smallest elements using PriorityQueue
46. Use Collections.unmodifiableList to create a read-only list
47. Make a thread-safe list using Collections.synchronizedList
48. Create a LRU Cache using LinkedHashMap
49. Group anagrams using Map and List
50. Implement a frequency counter using Map with computeIfAbsent()

Recursion

1. Print numbers from 1 to N
2. Print numbers from N to 1
3. Print numbers from 1 to N (tail recursion)
4. Print numbers from N to 1 (tail recursion)
5. Sum of first N natural numbers
6. Factorial of a number
7. Power of a number (a^b)
8. Print Fibonacci series up to N terms

9. Find nth Fibonacci number
10. Check if a string is palindrome
11. Reverse a string using recursion
12. Count digits of a number
13. Sum of digits of a number
14. Product of digits of a number
15. Check if a number is palindrome
16. Find GCD of two numbers
17. Find LCM using recursion
18. Convert decimal to binary
19. Convert binary to decimal
20. Print all subsets of a string
21. Generate all binary strings of length N
22. Generate all permutations of a string
23. Generate all combinations of balanced parentheses
24. Tower of Hanoi
25. Find maximum element in array
26. Find minimum element in array
27. Find sum of elements in array
28. Reverse an array using recursion
29. Check if array is sorted
30. Linear search using recursion
31. Binary search using recursion
32. Replace all 'pi' with '3.14' in a string
33. Remove all occurrences of 'x' from a string
34. Remove duplicates from string
35. Move all 'x' to the end of string
36. Merge sort using recursion
37. Quick sort using recursion
38. Insertion sort using recursion
39. Count paths in NxM grid (right & down only)
40. Climb stairs problem (1, 2 steps)
41. Count number of ways to reach target with dice rolls
42. Nth term of a custom-defined recurrence relation
43. Generate power set of a string
44. Solve Sudoku using recursion
45. Word Break Problem (recursive backtracking)
46. Generate combinations of k numbers out of n
47. Generate permutations with spaces
48. Find all subsets with sum K
49. Subset sum problem (basic recursion)
50. Count ways to partition array into equal sum subsets

LogicWhile

Address: MIG 87, First Floor, KPHB Phase I, Hyderabad – 500072
Mobile No: +91 89857 02244, 85898 02244.

LinkedList

1. Create and traverse a singly linked list
2. Insert node at the beginning
3. Insert node at the end
4. Insert node at a given position
5. Delete node from beginning
6. Delete node from end
7. Delete node at a given position
8. Search an element in linked list
9. Reverse a linked list (iterative)
10. Reverse a linked list (recursive)
11. Find length of a linked list
12. Find middle element of a linked list
13. Detect a loop in linked list (Floyd's cycle detection)
14. Remove loop from linked list
15. Find the starting node of the loop
16. Check if linked list is palindrome
17. Merge two sorted linked lists
18. Find intersection point of two linked lists
19. Remove duplicates from sorted linked list
20. Remove duplicates from unsorted linked list
21. Swap nodes in pairs
22. Segregate even and odd nodes
23. Add two numbers represented by linked lists
24. Sort a linked list (Merge Sort)
25. Flatten a multilevel linked list
26. Rotate linked list by K nodes
27. Delete N nodes after M nodes
28. Clone a linked list with random pointers
29. Convert binary number in linked list to integer
30. Intersection of two linked lists (using Hashing or Two Pointers)
31. Remove all occurrences of a given key in a linked list
32. Find the Nth node from the end of the list (two-pointer technique)
33. Check if two linked lists are identical
34. Reverse a doubly linked list
35. Convert a linked list to a binary tree (level-order)
36. Convert a binary tree to a doubly linked list (in-order traversal)
37. Reverse every K nodes in a linked list
38. Delete a node without head pointer
39. Rearrange linked list in zigzag fashion (L0→Ln→L1→Ln-1→…)
40. Partition linked list around a value (like QuickSort pivot)

LogicWhile

Address: MIG 87, First Floor, KPHB Phase I, Hyderabad – 500072
Mobile No: +91 89857 02244, 85898 02244.

Trees

1. Create a binary tree from user input
2. Preorder traversal (recursive)
3. Inorder traversal (recursive)
4. Postorder traversal (recursive)
5. Level order traversal (BFS using queue)
6. Preorder traversal (iterative)
7. Inorder traversal (iterative)
8. Postorder traversal (iterative with two stacks)
9. Count total nodes in a binary tree
10. Count leaf nodes in a binary tree
11. Count non-leaf (internal) nodes
12. Calculate height of a binary tree
13. Find diameter of binary tree (naive & optimized)
14. Check if two trees are identical
15. Check if a tree is height-balanced
16. Convert tree to mirror tree
17. Find Lowest Common Ancestor (LCA)
18. Find distance between two nodes
19. Print all nodes at distance K from root
20. Print all nodes at distance K from target node
21. Find sum of all nodes
22. Find max element in binary tree
23. Find min element in binary tree
24. Check if a path with given sum exists (root-to-leaf)
25. Print all root-to-leaf paths
26. Sum of all root-to-leaf paths
27. Zigzag (spiral) level order traversal
28. Boundary traversal of binary tree
29. Vertical order traversal
30. Top view of binary tree
31. Bottom view of binary tree
32. Right view of binary tree
33. Left view of binary tree
34. Serialize and deserialize a binary tree
35. Construct binary tree from inorder & preorder
36. Construct binary tree from inorder & postorder
37. Find ancestors of a given node
38. Check if tree is symmetric
39. Print vertical sum of a binary tree
40. Check if tree is a sum tree
41. Convert binary tree to doubly linked list (in-order)
42. Flatten binary tree to linked list (preorder)

43. Find maximum path sum from any node to any node
44. Print nodes without siblings
45. Find level with maximum sum
46. Find all nodes at a given depth
47. Count nodes in a complete binary tree (optimized)
48. Invert a binary tree
49. Check if a binary tree is a subtree of another
50. Find the kth smallest/largest element in a BST

Binary Search

1. Binary Search on sorted array (iterative)
2. Binary Search on sorted array (recursive)
3. First occurrence of a target in sorted array
4. Last occurrence of a target in sorted array
5. Count occurrences of a target element
6. Find element in rotated sorted array
7. Find minimum element in rotated sorted array
8. Find peak element in an array
9. Square root of a number using binary search
10. Cube root of a number using binary search
11. Binary search in infinite sorted array
12. Find insertion position (lower bound)
13. Find smallest letter greater than target
14. First bad version
15. Search in 2D matrix (matrix is row and column sorted)
16. Median of two sorted arrays
17. Kth smallest element in sorted matrix
18. Aggressive cows problem (maximum minimum distance)
19. Book allocation problem (minimize max pages)
20. Painter's partition problem
21. Split array largest sum (binary search on answer)
22. Koko eating bananas (minimum eating speed)
23. Ship packages within D days
24. Search in bitonic array
25. Search in mountain array

Dynamic Programming

1. Fibonacci number (recursion + memoization + tabulation)
2. Climbing Stairs (ways to reach Nth step)
3. Count ways to reach Nth stair (1 or 2 steps)
4. Minimum cost to reach end (stair cost)
5. Minimum number of jumps to reach end
6. Maximum sum of non-adjacent elements

7. House Robber (non-adjacent sum problem)
8. Count ways to cover distance (1, 2, 3 steps)
9. Decode Ways (1–26 mapping to A–Z)
10. 0/1 Knapsack (basic version)
11. Subset Sum Problem
12. Count subsets with given sum
13. Partition Equal Subset Sum
14. Coin Change – Minimum coins
15. Coin Change – Number of ways
16. Longest Common Subsequence (LCS)
17. Longest Palindromic Subsequence
18. Longest Palindromic Substring
19. Longest Increasing Subsequence (LIS)
20. Maximum sum increasing subsequence
21. Number of ways to reach cell
22. Unique Paths with obstacles
23. Maximum path sum in a matrix (top to bottom)
24. Count binary strings without consecutive 1's
25. Catalan Number (nth value using DP)