MASTER YOUR LOGIC BUILDING
(BEFORE STARTING DSA)

Phase 1 – Conditional Thinking (If–Else, Boolean Logic)
Phase 2 – Looping & Patterns (Iteration & Flow)
Phase 3 – Recursion (Thinking in self- reference)
Phase 4 – Basic Arrays(Iterative Logical Thinking)
Phase 5 – Strings (Basic Logic Building)
Phase 6 – Mixed Logical Challenges (Applied Reasoning)

Phase 1 – Conditional Thinking (If–Else, Boolean Logic)

Goal: Understand how to make decisions using conditions.
Topics covered: Relational operators, logical operators, nested if, multiple conditions.
Target Questions: 50

Level 1: Simple Conditions (Getting started)

1. Take a number and print whether it's positive, negative, or zero.
2. Check if a number is even or odd.
3. Check if a number is divisible by 5.
4. Check if a number is divisible by both 3 and 5.
5. Check if a given year is a leap year.
6. Take two numbers and print the larger one.
7. Take three numbers and print the largest.
8. Take a temperature value and print "Cold", "Warm", or "Hot" using range conditions.
9. Take a character and check if it's a vowel or consonant.
10. Take a character and check whether it's uppercase, lowercase, a digit, or a special character.

Level 2: Nested If & Multiple Conditions

1. Take three sides and check if they form a valid triangle.
2. If the sides form a valid triangle, determine whether it is equilateral, isosceles, or scalene.
3. Take marks (0–100) and print the corresponding grade (A/B/C/D/F).
4. Check if one of two given numbers is a multiple of the other.
5. Take the hour of the day (0–23) and print "Good Morning", "Good Afternoon", "Good Evening", or "Good Night".
6. Check voting eligibility for a given age (18+).
7. Take two numbers and determine whether both are even, both are odd, or one is even and one is odd.
8. Take an alphabet character and check if it lies between 'a' and 'm' or 'n' and 'z'.
9. Take a day number (1–7) and print the corresponding day name.
10. Take a month number (1–12) and print the number of days in that month (ignore leap years).

? Level 3: Math and Number Logic

1. Take a 3-digit number and check if all digits are distinct.
2. Take a 3-digit number and determine if the middle digit is the largest, smallest, or neither.
3. Take a 4-digit number and check if the first and last digits are equal.
4. Check whether a given integer is single-digit, double-digit, or multi-digit.
5. Check if a number is a multiple of 7 or ends with 7.
6. Take coordinates (x, y) and determine which quadrant the point lies in.
7. Check if an amount can be evenly divided into 2000, 500, and 100 currency notes.
8. Check if a number lies within the range [100, 999].
9. Take two angles of a triangle and compute the third angle.
10. Check whether a number is a perfect square (without using the square root function).

? Level 4: Logical Operators & Compound Statements

1. Take a character and check if it is a letter, a digit, or neither.
2. Take a number and print “Fizz” if divisible by 3, “Buzz” if divisible by 5, and “FizzBuzz” if divisible by both.
3. Take three numbers and print the median value (neither maximum nor minimum).
4. Take 24-hour time (hours and minutes) and print whether it is AM or PM.
5. Take income and age, and check if eligible for tax (age > 18 and income > 5 L).
6. Take two numbers and check if both are positive and their sum is less than 100.
7. Take a single digit (0–9) and print its word form (“Zero” to “Nine”).
8. Take a weekday number (1–7) and determine if it is a weekday or weekend.
9. Take electricity units consumed and calculate the bill as per slabs (using if-else).
10. Take a password string and check basic rules (length ≥ 8 and contains at least one digit).

● Level 5: Creative / Tricky Logical Scenarios

1. Take coordinates (x, y) and check if the point lies on the X-axis, Y-axis, or at the origin.
2. Take three numbers and check if they can form a Pythagorean triplet.
3. Take day and month and check if it forms a valid calendar date (ignoring leap years).
4. Take time (hours and minutes) and print the smaller angle between the hour and minute hands.
5. Take three numbers and check if they are in arithmetic progression.
6. Take three numbers and check if they are in geometric progression.
7. Take a 3-digit number and check if the sum of the first and last digit equals the middle digit.
8. Take an integer (1–9999) and check if the sum of its digits is greater than the product of its digits.
9. Take two dates (day and month) and determine which one comes first in the calendar.
10. Take a year and print the corresponding century (e.g., “19th century”, “20th century”)

PHASE 2 — LOOPING & PATTERNS (ITERATION & FLOW)

🎯 **Goal:** Master loops, iteration, and dry-run thinking.
Topics covered: for, while, nested loops, break/continue, mathematical series.
Target Questions: 40–50

🔹 Level 1: Basic Looping (Introductory)

1. Print numbers from 1 to 10.
2. Print all even numbers between 1 and 100.
3. Print all odd numbers between 1 and 100.
4. Print numbers from 10 down to 1.
5. Print the table of a given number (n × 1 to n × 10).
6. Print the sum of first n natural numbers.
7. Print the sum of all even numbers up to n.
8. Print the sum of all odd numbers up to n.
9. Print the factorial of a given number.
10. Print the product of digits of a given number.

🔹 Level 2: Number-based Looping Logic

1. Count the number of digits in a given number.
2. Print the reverse of a given number.
3. Check if a number is a palindrome.
4. Find the sum of digits of a number.
5. Check if a number is an Armstrong number.
6. Check if a number is a perfect number.
7. Print all prime numbers between 1 and 100.
8. Check if a number is prime or not.
9. Print Fibonacci series up to n terms.
10. Print sum of first n terms of Fibonacci series.

🔹 Level 3: Mathematical & Logical Patterns

1. Print the squares of numbers from 1 to n.
2. Print cubes of numbers from 1 to n.
3. Print all numbers between a and b divisible by 7.
4. Find HCF (GCD) of two numbers using loops.
5. Find LCM of two numbers using loops.
6. Print all factors of a given number.
7. Find the sum of all factors of a number.
8. Check if a number is a strong number (sum of factorials of digits = number).
9. Print first n terms of an arithmetic progression (a, d).
10. Print first n terms of a geometric progression (a, r).

⚫ Level 4: Pattern Printing (Stars & Numbers)

(These train nested loop thinking — must-do for logic growth.)
Solve this Sheet for Star printing Practice: STRONG YOUR LOGIC BUILDING.PDF

● Level 5: Logical Loop Combinations

1. Print all numbers whose sum of digits is even (1–100).
2. Count how many numbers between 1–500 are divisible by 7 but not by 5.
3. Print all numbers that are palindromes between 1–500.
4. Print numbers between 1–100 whose digits add up to a multiple of 3.
5. Find the smallest and largest digit in a given number.
6. Print all numbers from 1–n whose binary representation has an even number of 1s.
7. Print a pattern where each row i prints i*i.
8. Print factorial of each number from 1 to n.
9. Print the sum of all odd digits and even digits separately in a number.
10. Take 5 numbers as input. If the user enters 0, skip it using continue. At the end, print the sum of all non-zero numbers entered.

PHASE 3 — RECURSION (THINKING IN SELF-REFERENCE)

Goal: Develop logical decomposition and base-condition thinking.
Topics covered: recursive definition, base cases, call stack tracing.
Target Questions: 30–40

Level 1: Foundation of Recursion (Base + Recursive Case)

1. Print numbers from 1 to n using recursion.
2. Print numbers from n down to 1 using recursion.
3. Print only even numbers from 1 to n recursively.
4. Print only odd numbers from 1 to n recursively.
5. Print sum of first n natural numbers recursively.
6. Print factorial of a number recursively.
7. Calculate power of a number (xn) using recursion.
8. Find nth Fibonacci number recursively.
9. Print Fibonacci series up to n terms recursively.
10. Find sum of digits of a number recursively.

Level 2: Number-based Recursive Thinking

1. Count the number of digits in a number recursively.
2. Reverse a number recursively.
3. Check if a number is a palindrome using recursion.
4. Find product of digits of a number recursively.
5. Find GCD (HCF) of two numbers using Euclid's algorithm recursively.
6. Convert a number to binary recursively.
7. Print digits of a number in words recursively (e.g., 123 → "one two three").
8. Calculate the sum of first n even numbers recursively.
9. Calculate the sum of first n odd numbers recursively.
10. Find nCr (Combination formula) recursively using Pascal's relation.

▯ Level 3: Pattern & Printing Problems

1. Print a line of n stars recursively.
2. Print a square of stars recursively (n×n).
3. Print a triangle of stars recursively (top-down).
4. Print a triangle of stars recursively (bottom-up).
5. Print pattern of numbers recursively (1 to n each row).
6. Print reverse triangle pattern recursively.
7. Print multiplication table of n recursively.
8. Print numbers in increasing and decreasing order in same function.
9. Print sum of series 1 + 2 + 3 + ... + n recursively and display each step.
10. Print pattern of characters (A, AB, ABC, ...) recursively.

● Level 4: String-based Recursion

1. Reverse a string using recursion.
2. Check if a string is palindrome using recursion.
3. Count vowels in a string recursively.
4. Remove all spaces from a string recursively.
5. Replace all occurrences of a character (say 'a' → 'x') recursively.
6. Remove all occurrences of a character from a string recursively.
7. Print all characters of a string one by one recursively.
8. Print the string in reverse order recursively (without using loops).
9. Convert a string to uppercase recursively.
10. Count consonants and vowels separately using recursion.

PHASE 4 — BASIC ARRAYS (Iterative Logical Thinking)

🎯 **Goal:** Build the ability to handle a collection logically.
Topics covered: traversal, frequency, simple manipulation, aggregations.
Target Questions: 30–40

▯ Level 1: Fundamentals of Arrays

1. Input n and take n integers into an array; print them.
2. Find the sum of all elements in an array.
3. Find the average of array elements.
4. Find the maximum element in an array.
5. Find the minimum element in an array.
6. Count how many elements are positive, negative, or zero.
7. Count how many elements are even and odd.
8. Find the index of the maximum element.
9. Find the index of the minimum element.
10. Take n elements and print only those greater than a given value k.

🔍 Level 2: Searching & Counting Logic

1. Input an element x — check if it exists in the array.
2. Count how many times a given element appears.
3. Find the first occurrence of a given number.
4. Find the last occurrence of a given number.
5. Check if all elements in an array are unique.
6. Find the sum of even elements only.
7. Find the sum of odd elements only.
8. Find the count of prime numbers in the array.
9. Count how many numbers are divisible by 3 and 5 both.
10. Count how many elements are perfect squares.

🔁 Level 3: Transformation & Manipulation

1. Create a new array containing squares of all numbers.
2. Create a new array containing only even elements.
3. Replace every negative number with 0.
4. Replace all even numbers with 1 and all odd with 0.
5. Swap the first and last elements of the array.
6. Reverse an array (without using built-in reverse).
7. Rotate an array by one position to the left.
8. Rotate an array by one position to the right.
9. Swap alternate elements (1st ↔ 2nd, 3rd ↔ 4th, etc.).
10. Copy one array to another manually.

⚫ Level 4: Aggregate & Comparative Thinking

1. Compare two arrays — check if they are equal (same elements & order).
2. Compare two arrays — check if they contain the same elements (ignore order).
3. Merge two arrays into a third array.
4. Find the common elements between two arrays.
5. Find elements that are in one array but not in the other.
6. Count how many elements are common between two arrays.
7. Find element-wise sum of two arrays (A[i] + B[i]).
8. Find element-wise product of two arrays.
9. Create a frequency array of numbers (count occurrence of each number).
10. Print all elements that appear more than once.

● Level 5: Logical & Applied Array Problems

1. Check if the array is sorted in ascending order.
2. Check if the array is sorted in descending order.
3. Find the second largest element in an array.
4. Find the second smallest element in an array.
5. Find the difference between the largest and smallest element.
6. Find the sum of all elements except the largest and smallest.
7. Count how many pairs of elements have a sum equal to a given number k.
8. Count how many elements are greater than the average of the array.
9. Print the frequency of each distinct element.
10. Print all unique elements (those that occur exactly once).

Phase 5 – Strings (Basic Logic Building – 50 Questions)

Category 1: Basic String Handling (10 Qs)

1. Take a string input and print its length.
2. Print the first and last character of a string.
3. Convert all characters of a string to uppercase.
4. Convert all characters of a string to lowercase.
5. Count how many characters (excluding spaces) are in the string.
6. Count how many words are in a sentence.
7. Take two strings and print them concatenated.
8. Compare two strings lexicographically (like dictionary order).
9. Print the ASCII value of each character in a string.
10. Check whether the string is empty or not.

Category 2: Counting & Character Analysis (10 Qs)

1. Count how many vowels and consonants are in a string.
2. Count the number of digits, letters, and special characters in a string.
3. Count how many uppercase and lowercase letters a string has.
4. Find the frequency of each character in a string (without using a map).
5. Count how many spaces are there in a sentence.
6. Count how many times a given character appears in a string.
7. Count how many alphabets are before ‘m’ and after ‘m’ in a given string.
8. Count how many substrings start and end with the same character (simple logic).
9. Print how many words start with a vowel in a sentence.
10. Count how many words end with ‘s’.

Category 3: Reversing & Palindromic Thinking (10 Qs)

1. Reverse a string without using built-in reverse.
2. Reverse each word in a sentence.
3. Reverse the order of words in a sentence.
4. Check whether a string is a palindrome.
5. Check if two strings are the reverse of each other.
6. Print the middle character(s) of a string.
7. Print the second half of the string in reverse.
8. Remove the first and last character and print the remaining string.
9. Reverse only characters, keeping digits in place.
10. Reverse string but skip spaces.

Category 4: Character & Word Manipulation (10 Qs)

1. Remove all vowels from a string.
2. Remove all spaces from a string.
3. Replace all vowels with '*'.
4. Replace all spaces with '_'.
5. Print the string after removing all digits.
6. Remove duplicate characters from a string.
7. Keep only the first occurrence of each character.
8. Remove consecutive duplicate characters (e.g., "aaabb" → "ab").
9. Swap case: uppercase → lowercase and lowercase → uppercase.
10. Shift each character by 1 (e.g., "abc" → "bcd").

Category 5: Word-level Thinking (10 Qs)

1. Print each word of a sentence on a new line.
2. Count how many words have even length.
3. Find the longest word in a sentence.
4. Find the shortest word in a sentence.
5. Swap first and last words in a sentence.
6. Print all words that start and end with the same letter.
7. Count how many words contain the letter 'a'.
8. Capitalize the first letter of each word.
9. Print the sentence in title case (first letter capital, rest lowercase).
10. Remove extra spaces between words (normalize spacing).

Phase 6 – Mixed Logical Challenges

🎯 **Goal:** Strengthen logical thinking with character manipulation.
Topics covered: char array logic, string length, substring, conditions.
Target Questions: 30–40

Category 1: Number-Based Logical Combinations (10 Qs)

1. Print all numbers between 1 and N that are divisible by both 3 and 5.
2. Find the sum of digits of a number (use loop).
3. Check if a number is an Armstrong number.
4. Print all Armstrong numbers between 1 and 1000.
5. Find the factorial of a number using recursion.
6. Count how many even digits a number contains.
7. Print all prime numbers between 1 and N.
8. Print the reverse of a number (123 → 321).
9. Check if a number is palindrome (121 → true).
10. Check if a number is perfect (sum of factors equals number).

Category 2: String + Logic Mix (10 Qs)

1. Check if two strings are anagrams (without using collections).
2. Count vowels in each word of a sentence.
3. Reverse words in a string if their length is even.
4. Replace every vowel in a string with its position (a=1, e=2...).
5. Print characters that appear more than once (without map).
6. Count words that start and end with the same letter.
7. Toggle case for every alternate word in a sentence.
8. Check if two strings are rotations of each other.
9. Find the word with maximum vowels in a sentence.
10. Remove duplicate words from a sentence.

Category 3: Array + Looping Logic (10 Qs)

1. Find the maximum and minimum element in an array.
2. Count how many positive, negative, and zero elements are in an array.
3. Print all unique elements from an array.
4. Reverse an array in-place.
5. Shift all zeros to the end of the array.
6. Count how many elements are even at an even index.
7. Merge two arrays into one.
8. Find the second largest element in an array.
9. Rotate an array by one position to the right.
10. Find the sum of all elements at odd indices.

Category 4: Nested Logic & Pattern Flow (10 Qs)

1. Print a multiplication table in a formatted grid (10x10).
2. Print all pairs in an array whose sum equals a given number.
3. Print all subarrays of a given array.
4. Check if an array is sorted (ascending or descending).
5. Count how many times a number appears consecutively in an array.
6. Find all pairs of characters in a string that are the same (nested loop).
7. Print pattern of increasing characters (A, AB, ABC...).
8. Print Pascal's triangle up to N rows.
9. Generate Fibonacci series up to N using recursion.
10. Print numbers in a spiral-like pattern (conceptual dry run).

Category 5: Applied Reasoning & Real-Life Logic (10 Qs)

1. Given marks of students, find how many passed (>= 40).
2. Take age inputs and count how many are adults, minors, seniors.
3. Validate a password (at least one uppercase, lowercase, digit, special char).
4. Simulate a simple calculator using switch-case.
5. Count how many times a coin lands on heads/tails (use random).
6. Print frequency of each digit in a number.
7. Find common elements between two arrays.
8. Print characters that are common in two strings.
9. Count how many prime numbers are there in an array.
10. Print all palindromic words from a sentence.