# Short Coding Problems

Duration – 2 sessions (90 min + 60 min)
Goal – Train students to explain **logic + flow**, not syntax.

## SECTION 1 – Array / List Basics

| No | Problem | Focus |
|---|---|---|
| 1 | Find largest element in array | Traversal, comparison |
| 2 | Find second largest element (no sort) | Two variable logic |
| 3 | Find smallest & second smallest | Nested check |
| 4 | Reverse an array | Swapping indices |
| 5 | Calculate sum, average | Basic arithmetic |
| 6 | Count even & odd numbers | Condition check |
| 7 | Count positive, negative, zero | Branching |
| 8 | Find duplicates in array | Frequency logic |
| 9 | Remove duplicates | Set or manual loop |
| 10 | Find missing number (1–n) | Sum formula vs actual sum |
| 11 | Sort array ascending / descending | Bubble / built-in |
| 12 | Merge two sorted arrays | Two-pointer approach |
| 13 | Rotate array left / right by K | Modulus logic |
| 14 | Find pair with given sum | Nested loop / hash set |
| 15 | Find intersection of two arrays | Logic & sets |

## SECTION 2 – String Operations

| No | Problem | Focus |
|---|---|---|
| 16 | Reverse string | Slice / loop |
| 17 | Palindrome check | Compare reverse |
| 18 | Count vowels & consonants | Membership test |
| 19 | Count words in sentence | Split logic |
| 20 | Find frequency of each char | Dict usage |
| 21 | Replace a substring | replace() or manual loop |
| 22 | Remove spaces / special chars | Filtering |
| 23 | Convert to title / toggle case | String methods |
| 24 | Check anagram (two words) | Sort / count |
| 25 | Find first non-repeated char | Counting order |
| 26 | Longest word in sentence | Split + len() |
| 27 | Count digits, letters, symbols | ASCII classification |
| 28 | Compress string (“aaabb”→“a3b2”) | Logic building |

## SECTION 3 – Mathematical / Number Logic

| No | Problem | Focus |
|---|---|---|
| 29 | Prime check | Loop / flag |
| 30 | Print all primes ≤ N | Nested loop |
| 31 | Factorial of number | Iterative / recursive |
| 32 | Fibonacci series | Loop / sequence |
| 33 | Sum of digits | Modulus loop |
| 34 | Armstrong number | Power of digits |
| 35 | Perfect number | Sum of divisors |
| 36 | Strong number | Factorial of digits |
| 37 | GCD & LCM | Euclidean method |
| 38 | Reverse digits of number | Arithmetic ops |
| 39 | Check palindrome number | Compare reverse |
| 40 | Convert decimal↔binary | Logic / format() |
| 41 | Count digits in number | Division loop |
| 42 | Sum of even / odd digits | Conditional |
| 43 | Power without operator | Loop multiply |

## SECTION 4 – Logic Tracing / Dry-Run / Pattern

| No | Problem | Focus |
|---|---|---|
| 44 | Predict output of small loop (sum of squares) | Iteration analysis |
| 45 | Count iterations in nested loop | Loop tracing |
| 46 | Identify logical error (range misuse, variable init) | Debugging |
| 47 | Pattern: right-triangle stars | Nested loops |
| 48 | Pattern: inverted triangle | Reverse iteration |
| 49 | Pattern: numeric triangle | Printing values |
| 50 | Predict final value after swaps / updates | Variable flow |

## SECTION 5 – Mixed / Real-World Mini-Tasks

| No | Problem | Focus |
|---|---|---|
| 51 | Calculate total bill from list of item prices | List sum |
| 52 | Validate email format | String conditions |
| 53 | Find highest scored student from dict | Dict traversal |
| 54 | Remove duplicates from customer order IDs | Set logic |
| 55 | Count frequency of each product sold | Dict / Counter |
| 56 | Check if two lists are identical (any order) | sort + == |
| 57 | Find common customers in two files (lists) | Intersection |
| 58 | Generate OTP of 6 digits | Random module |
| 59 | Sum of matrix diagonals | 2D array traversal |
| 60 | Transpose a matrix | Nested loops / zip |

## Cognizant SME Evaluation Focus

| Skill | What They Observe |
|---|---|
| Logic Building | Step-wise approach |
| Code Clarity | Clean indentation, comments |
| Output Accuracy | Matching dry-run |
| Confidence | Calm explanation |
| Optimization | Avoid unnecessary loops |