

Capgemini DSA Previous Year Questions

Easy

- Write the code for this: Check if a year is a leap year.
(Given a year, determine whether it is a leap year following Gregorian rules.)
- Write the code for this: Find the GCD (Greatest Common Divisor) of two numbers.
(Return the largest integer that divides both given numbers without remainder.)
- Write the code for this: Reverse an array.
(Given an array, return it with elements in reverse order.)
- Write the code for this: Find the factorial of a large number.
(Compute $n!$ for large n — handle big results using big integer arithmetic or strings.)
- Write the code for this: Check if a number is a perfect square.
(Decide whether a non-negative integer is the square of some integer.)
- Write the code for this: Concatenate two linked lists.
(Given two singly linked lists, join them end-to-end and return the head of the resulting list.)
- Write the code for this: Find the number of nodes in a linked list.
(Return the count of nodes in the given singly linked list.)
- Write the code for this: Implement a stack using a linked list.
(Build stack operations (push, pop, peek, isEmpty) backed by a linked list.)
- Write the code for this: Implement a queue using a linked list.
(Implement enqueue, dequeue, peek, and isEmpty using a linked list.)

- Write the code for this: Find the maximum depth of a binary tree.
(Return the number of levels in a binary tree — the longest root→leaf path length.)
 - Write the code for this: Convert a decimal number to binary.
(Return the binary string representation of a non-negative integer.)
 - Write the code for this: Find the element that appears only once in an array where every other element appears twice.
(Given an array where every value appears twice except one, return the unique value.)
 - Write the code for this: Sort a string of characters.
(Return the characters of a string sorted in ascending order.)
 - Write the code for this: Capitalize the first letter of each word in a sentence.
(Given a sentence, return it with every word's first character capitalized.)
 - Write the code for this: Count the number of set bits in an integer.
(Return how many 1 bits are in the binary representation of an integer.)
 - Write the code for this: Find the power of a number using recursion.
(Compute $\text{base}^{\text{exponent}}$ recursively (for non-negative integer exponents).)
 - Write the code for this: Search for a node in a Binary Search Tree (BST).
(Given the root of a BST and a value, return the node containing that value if present.)
 - Write the code for this: Insert a node into a BST.
(Insert a value into a BST while preserving BST order and return the root.)
 - Write the code for this: Find the minimum value node in a BST.
(Return the node with the smallest key (leftmost node) in the BST.)
 - Write the code for this: Check if a given key exists in a hash map.
(Given a hash map and a key, return whether the key is present.)
-

Medium

- Write the code for this: Find the pairs in an array with a given sum (Two Sum problem).
(Return all index pairs or values whose sum equals the given target.)
- Write the code for this: Find the subarray with the largest sum (Kadane's Algorithm).
(Return the maximum sum of any contiguous subarray.)
- Write the code for this: Print a given matrix in spiral form.
(Traverse a 2D matrix in clockwise spiral order and return the sequence of elements.)
- Write the code for this: Find the triplet that sums to a given value.
(Find three numbers in an array whose sum equals a given target; return indices or values.)
- Write the code for this: Detect and remove a loop in a linked list.
(If a cycle exists, remove it so the list becomes linear and return the head.)
- Write the code for this: Add two numbers represented by linked lists.
(Given two lists with digits in reverse order, return their sum as a linked list.)
- Write the code for this: Reverse a linked list in groups of a given size K.
(Reverse every consecutive block of k nodes in the list; leftover nodes remain as is.)
- Write the code for this: Print the boundary traversal of a binary tree.
(Output nodes on the boundary: left boundary, leaves, then right boundary in order.)
- Write the code for this: Check if a binary tree is balanced.
(Determine whether the difference of heights of left and right subtrees is ≤ 1 for every node.)

- Write the code for this: Convert a sorted array to a balanced BST. (Build a height-balanced BST from a sorted array (median as root recursively).)
- Write the code for this: Find the shortest path between two nodes in an unweighted graph using BFS. (Use BFS to return the minimum number of edges from source to target or the path itself.)
- Write the code for this: Detect a cycle in a directed graph using DFS. (Return whether a directed graph contains a cycle using depth-first search and recursion stack.)
- Write the code for this: Find the edit distance between two strings. (Return the minimum number of insert/delete/replace operations to convert one string to the other.)
- Write the code for this: Find the next lexicographically greater permutation of a given sequence of numbers. (Transform the sequence to its next permutation in lexicographic order, or return the lowest order if none.)
- Write the code for this: Evaluate a postfix expression using a stack. (Compute the value of an expression given in Reverse Polish Notation.)
- Write the code for this: Find all subsets of a given set (powerset). (Return all possible subsets (the powerset) of a given set or array.)
- Write the code for this: Find the ugly numbers up to a given N. (Generate numbers whose only prime factors are 2, 3, and 5 up to the Nth such number.)
- Write the code for this: Implement a min-heap using an array. (Implement heap operations (push, pop, peek) using an array-based binary heap for min ordering.)
- Write the code for this: Find the square root of a number using binary search. (Return $\text{floor}(\sqrt{x})$ by binary searching the integer root.)
- Write the code for this: Segregate even and odd nodes in a linked list. (Reorder a linked list so that all even-valued nodes come before odd-valued nodes while preserving relative order.)

Hard

- Write the code for this: Find the median from a data stream.
(Design a structure that supports adding numbers and returning the median at any time efficiently.)
- Write the code for this: Implement word ladder problem (shortest transformation sequence from one word to another).
(Given begin, end and a word list, return the shortest sequence of single-letter transformations from begin to end.)
- Write the code for this: Find the smallest window in a string containing all characters of another string.
(Return the shortest substring of S that contains all characters of T including multiplicity.)
- Write the code for this: Find the maximum of all subarrays of size K.
(For each sliding window of length k, compute the maximum element efficiently.)
- Write the code for this: Find the optimal strategy for a game.
(Given a two-player perfect-information game (e.g., optimal coin-picking), compute the strategy maximizing the player's outcome.)
- Write the code for this: Find the minimum number of platforms required for a railway station.
(Given arrival/departure times, compute the minimum number of platforms so no train waits.)
- Write the code for this: Clone a linked list with next and random pointers.

(Create a deep copy of a list where each node has next and random references.)

- Write the code for this: Find the egg dropping puzzle.
(Determine the minimum number of trials in worst case to find highest floor from which an egg won't break using given eggs and floors.)
 - Write the code for this: Find the longest valid parentheses substring.
(Given a parentheses string, return the length of the longest valid (well-formed) parentheses substring.)
 - Write the code for this: Implement merge sort for linked lists.
(Sort a singly linked list in $O(n \log n)$ time using merge sort.)
 - Write the code for this: Find the alien dictionary order in a given sorted dictionary of an alien language.
(Given words sorted in an unknown language, deduce a valid character ordering that explains the sorting.)
 - Write the code for this: Find the maximum rectangular area in a binary matrix.
(Given a binary matrix, find the largest rectangle containing only 1's and return its area.)
-