**.**

**🔹Here’s a step-by-step set of array questions (in increasing order of difficulty). If you solve these, you’ll be well-prepared for interviews:**

**Basic Level (must master first)**

1. Find the **largest** and **smallest** element in an array.
2. Find the **sum** and **average** of elements in an array.
3. Count **even and odd** numbers in an array.
4. Search an element in an array (**Linear Search**).
5. Find the **second largest** and **second smallest** element.
6. Find the **maximum and minimum difference** between elements.
7. Count frequency of each element in the array.
8. Check if the array is **sorted** (ascending or descending).
9. Remove duplicates from an array (without using Set).

**🔹 Intermediate Level (commonly asked in interviews)**

1. Rotate an array **left/right by k positions**.
2. Find the **missing number** in an array from 1 to n.
3. Find the **duplicate number** in an array.
4. Find the element that occurs **only once** when every other element occurs twice.
5. Merge two sorted arrays into one sorted array.
6. Find the **intersection** and **union** of two arrays.
7. Move all **zeros to the end** of the array without changing order.
8. Find the **majority element** (element occurring > n/2 times).
9. Find **subarray with given sum** (positive numbers).
10. Find the **maximum product of two numbers** in an array.

**🔹 Medium Level (slightly tricky but important for service-based)**

1. Kadane’s Algorithm → Find **maximum subarray sum** (very common).
2. Find the **longest increasing subsequence** (basic DP intro).
3. Find the **peak element** (element greater than its neighbors).
4. Find the **equilibrium index** (sum of left part = sum of right part).
5. Find the **leaders in an array** (element greater than all elements to its right).
6. Find the **pair of elements whose sum is closest to a given target**.
7. Find all pairs in array with a **given sum**.
8. Sort an array of **0s, 1s, and 2s** without using sorting algorithm (Dutch National Flag).
9. Find the **longest subarray with equal number of 0s and 1s**.
10. Find the **maximum difference between two elements** such that larger element appears after smaller element.

✅ If you master these 30 problems, you’ll be well-prepared for **Cognizant, TCS, Infosys, Wipro, Accenture, Capgemini**, and other service-based companies.  
✅ Most of them are **array + logic**, no heavy DSA required.

* Great that you’re targeting **Cognizant on-campus service-based placement**! 🚀  
  Cognizant (CTS) usually asks **basic to medium-level coding questions** in interviews/assessments, mainly focused on:
* Arrays
* Strings
* Searching & Sorting
* Recursion (basic)
* Pattern printing
* LinkedList (rare for service-based, but possible)
* Stack/Queue (basic)
* Simple problem-solving (counting, mathematics, conversions)

Here’s a **comprehensive list of coding questions Cognizant typically asks** (from recent drives and practice sets):

**✅ Arrays**

1. Find the largest and smallest element in an array.
2. Find the second largest/second smallest element.
3. Reverse an array (without extra space).
4. Rotate an array (left/right by k positions).
5. Find the missing number in an array (1 to n).
6. Find duplicate elements in an array.
7. Find the frequency of each element in an array.
8. Merge two sorted arrays.
9. Find the subarray with the maximum sum (Kadane’s Algorithm).
10. Count pairs with a given sum.

**✅ Strings**

1. Reverse a string without using extra string functions.
2. Check if a string is a palindrome.
3. Find the frequency of characters in a string.
4. Remove duplicate characters from a string.
5. Find the first non-repeating character.
6. Check if two strings are anagrams.
7. Count vowels, consonants, digits, and special characters.
8. Convert string to uppercase/lowercase without library functions.
9. Replace spaces with “%20” (URLify problem).
10. Check whether two strings are rotations of each other.

**✅ Mathematics / Number-based**

1. Check if a number is prime.
2. Find factorial (using recursion & iteration).
3. Find Fibonacci series up to n terms.
4. Reverse a number.
5. Check if a number is palindrome.
6. Find GCD and LCM.
7. Armstrong number check.
8. Strong number check.
9. Count digits in a number.
10. Find sum of digits until a single digit remains.

**✅ Pattern Printing (very common in service-based)**

1. Half pyramid / inverted pyramid of stars.
2. Number pyramid.
3. Floyd’s triangle.
4. Pascal’s triangle.
5. Diamond pattern.

**✅ Searching and Sorting**

1. Linear Search.
2. Binary Search.
3. Bubble Sort.
4. Selection Sort.
5. Insertion Sort.
6. Merge Sort (rare for service-based, but sometimes asked).

**✅ Linked List (basic)**

1. Insert a node at beginning/end.
2. Delete a node.
3. Reverse a linked list.

**✅ Stack / Queue**

1. Implement stack using array.
2. Implement queue using array.
3. Balanced parentheses check using stack.

**✅ Miscellaneous**

1. Find the count of words in a sentence.
2. Find the maximum occurring character in a string.
3. Count pairs with difference k in an array.
4. Find leaders in an array.
5. Print prime numbers in a given range.

👉 In **Cognizant GenC/GenC Next**, they usually ask **2 coding questions** in the online test:

* One easy (array/string/basic math).
* One medium (string manipulation/array problem).

Would you like me to **write full Java solutions with explanations** for each of these (like a preparation sheet), step by step — starting from Arrays → Strings → Patterns → Others — so you can practice them for Cognizant?

**You said:**

yes

**ChatGPT said:**

Perfect 👌 Since you’re targeting **Cognizant service-based company (on-campus)**, I’ll build you a **step-by-step coding preparation sheet** in **Java** (with full code + explanation).

We’ll go **topic by topic** (Arrays → Strings → Patterns → Math → Others).  
This way you can master all the **possible Cognizant interview coding questions**.