# ARRAY DSA Questions   EASY -MEDIUM

| | COMPANY | QUESTION NAME | APPROACH | GFG | LEETCODE | NOTES |
|---|---|---|---|---|---|---|
| 1 | *Infosys*, *Oracle Wipro, Morgan Stanley* | Largest Element in Array<br>TC: O(n)  SC : O(1) | | GFG | | If n is not initialized, u r using the size of the vector &arr, you must explicitly initialize `int n = arr.size();` |
| 2 | *SAP Labs,Rockstand* | Second Largest elem in array without sorting<br>TC: O(n)  SC : O(1) | | GFG | | Visible array→ `arr[0]`. Invisible data→ `INT_MIN`<br>`Can't trust the data→ INT_MAX`<br>`INT_MIN → start with the` **worst max**<br>`INT_MAX → start with the` **worst min** |
| 3 | | Check if array is sorted<br>TC: O(n)  SC : O(1) | | GFG | | |
| 4 | *Zoho*, *Morgan Stanley*, *Microsoft*, *Samsung*, *Google*, *Wipro*, *Xome* | Remove Duplicates Sorted Array<br><br>TC: O(n)  SC : O(1) | *2 pointers* | GFG | LEETCODE | |
| 5 | | Rotate Array by One(clockwise direction/right)<br>TC: O(n)  SC : O(1) | | GFG | | Whenever they say rotate left/right, look at the key elem that moves, either first/last & shift accordingly.<br>Right Rotation by 1→**last element** (`arr[n-1]`)<br>Left Rotation by 1→**first element** (`arr[0]`)<br>Dont focus on shifting all elements. |
| 6 | *Amazon*, *Microsoft*, *MAQ Software* | Rotate the array to the left (counter-clockwise)<br><br>rotate the array to the right | | GFG | LEETCODE | **Left = F-R-A** → First–Rest–All<br>**Right = A-F-R** → All–First–Rest |
| 7 | *Paytm,Amazon Microsoft*, *Samsung*,  *SAP Labs, Linkedin*, *Bloomberg* | Move All Zeroes to End<br>TC: O(n)  SC : O(1) | *2 pointers* | GFG | LEETCODE | |
| 8 | *Amazon* | Union of 2 Sorted Arrays<br>TC: O(n1 +n2)<br>SC :O(n1+n2) WC | *2 pointers* | GFG | | |
| 9 | | Intersection of two sorted arrays<br>TC: O(n1 +n2)<br>SC :O(n1+n2) WC<br>Intersection of two unsorted arrays (LC) | *2 pointers* | GFG | LEETCODE | Intersection of two unsorted arrays (LC)<br>We used unordered map |
| 10 | *Flipkart, Morgan Stanley, Accolite, D-E-Shaw, OlaCabs, Pay, Visa, Intuit, Adobe, CISCO, Qualcomm TCS* | Missing number | *XOR* | GFG | LEETCODE | *Range Type:* `1 to n` *(like GFG)*<br>*→ XOR loop range: from 1 to n-1*<br>*→* ✅ *Yes, you need to add* `xor1 ^= n` *outside the loop to include the last number.*<br><br>*Range Type:* `0 to n` *(like Leetcode)*<br>*→ XOR loop range: from 0 to n*<br>*→* ❌ *No extra step needed — the loop already covers the full range.* |

| | | | | | | |
|---|---|---|---|---|---|---|
| 11 | *Amazon, Google ,META* | Max Consecutive 1's | | | LEETCODE | |
| 12 | *Amazon* | Single Number | | GFG | LEETCODE | |
| 13 | *Zoho, Flipkart, Morgan Stanley, Accolite, Amazon, Microsoft, FactSet, Hike, Adobe, Google, Wipro, SAP Labs, CarWale* | Two Sum - Pair with Given Sum | *2 pointers, HASHMAP* | GFG | LEETCODE | **Return Indices** → *Use HashMap (LeetCode style)*<br><br>**Return True/False** → *Use HashMap or Two Pointers if sorted (GFG style* |
| | | | | GFG | LEETCODE | |
| | | | | GFG | LEETCODE | |
| | | | | GFG | LEETCODE | |
| | | | | GFG | LEETCODE | |
| | | | | GFG | LEETCODE | |
| | | | | GFG | LEETCODE | |
| | | | | GFG | LEETCODE | |