🌟 20 Must-Know Array Questions in Java, C++, Python

✍️ BY: coding_error1

✅ 1. Find Maximum and Minimum in Array

🧠 *Find the smallest and largest element in the array.*

Java:

```
java
CopyEdit
int max = arr[0], min = arr[0];
for (int i = 1; i < arr.length; i++) {
    max = Math.max(max, arr[i]);
    min = Math.min(min, arr[i]);
}
```

C++:

```
cpp
CopyEdit
int max = arr[0], min = arr[0];
for (int i = 1; i < n; i++) {
    max = std::max(max, arr[i]);
    min = std::min(min, arr[i]);
}
```

Python:

```
python
CopyEdit
max_val = max(arr)
min_val = min(arr)
```

✅ 2. Reverse the Array

🧠 *Reverse elements in-place from start to end.*

Java:

```
java
CopyEdit
int start = 0, end = arr.length - 1;
while (start < end) {
    int temp = arr[start];
    arr[start++] = arr[end];
```

```
    arr[end--] = temp;
}
```

C++:

```
cpp
CopyEdit
int start = 0, end = n - 1;
while (start < end) {
    swap(arr[start++], arr[end--]);
}
```

Python:

```
python
CopyEdit
arr.reverse()
# or arr = arr[::-1]
```

✅ 3. Check if Array is Sorted

🧠 *Determine whether array elements are in non-decreasing order.*

Java:

```
java
CopyEdit
boolean sorted = true;
for (int i = 1; i < arr.length; i++) {
    if (arr[i] < arr[i - 1]) {
        sorted = false;
        break;
    }
}
```

C++:

```
cpp
CopyEdit
bool sorted = true;
for (int i = 1; i < n; i++) {
    if (arr[i] < arr[i - 1]) {
        sorted = false;
        break;
    }
}
```

Python:

```
python
CopyEdit
sorted = all(arr[i] <= arr[i+1] for i in range(len(arr)-1))
```

✅ 4. Kth Smallest & Largest Element

🧠 *Find the K-th smallest/largest using sorting.*

Java:

```
java
CopyEdit
Arrays.sort(arr);
int kthSmallest = arr[k - 1];
int kthLargest = arr[arr.length - k];
```

C++:

```
cpp
CopyEdit
sort(arr, arr + n);
int kthSmallest = arr[k - 1];
int kthLargest = arr[n - k];
```

Python:

```
python
CopyEdit
arr.sort()
kth_smallest = arr[k - 1]
kth_largest = arr[-k]
```

✅ 5. Left Rotate by One

🧠 *Move all elements left by one position, first becomes last.*

Java:

```
java
CopyEdit
int first = arr[0];
for (int i = 1; i < arr.length; i++)
    arr[i - 1] = arr[i];
arr[arr.length - 1] = first;
```

C++:

```
cpp
CopyEdit
int first = arr[0];
for (int i = 1; i < n; i++)
    arr[i - 1] = arr[i];
arr[n - 1] = first;
```

Python:

```
python
```

```
CopyEdit
first = arr.pop(0)
arr.append(first)
```

✅ 6. Move All Zeros to End

🧠 *Shift all non-zero elements to the left, keep relative order.*

Java:

```
java
CopyEdit
int index = 0;
for (int i = 0; i < arr.length; i++)
    if (arr[i] != 0) arr[index++] = arr[i];
while (index < arr.length) arr[index++] = 0;
```

C++:

```
cpp
CopyEdit
int index = 0;
for (int i = 0; i < n; i++)
    if (arr[i] != 0) arr[index++] = arr[i];
while (index < n) arr[index++] = 0;
```

Python:

```
python
CopyEdit
index = 0
for i in range(len(arr)):
    if arr[i] != 0:
        arr[index] = arr[i]
        index += 1
while index < len(arr):
    arr[index] = 0
    index += 1
```

✅ 7. Union of Two Arrays

🧠 *Get distinct elements from both arrays.*

Java:

```
java
CopyEdit
Set<Integer> union = new HashSet<>();
for (int x : arr1) union.add(x);
for (int x : arr2) union.add(x);
```

C++:

```
cpp
```

```
CopyEdit
set<int> unionSet;
for (int i = 0; i < n1; i++) unionSet.insert(arr1[i]);
for (int i = 0; i < n2; i++) unionSet.insert(arr2[i]);
```

Python:

```
python
CopyEdit
union = set(arr1) | set(arr2)
```

✅ 8. Missing Number in Range 1 to N

🧠 *Find missing number using sum formula.*

Java:

```
java
CopyEdit
int n = arr.length + 1;
int total = n * (n + 1) / 2;
for (int x : arr) total -= x;
```

C++:

```
cpp
CopyEdit
int total = (n + 1) * (n + 2) / 2;
for (int i = 0; i < n; i++) total -= arr[i];
```

Python:

```
python
CopyEdit
n = len(arr) + 1
missing = n * (n + 1) // 2 - sum(arr)
```

✅ 9. Find the Duplicate Number

🧠 *Use hashing or Floyd's cycle (not shown here).*

Java:

```
java
CopyEdit
Set<Integer> seen = new HashSet<>();
for (int x : arr) {
    if (!seen.add(x)) {
        System.out.println("Duplicate: " + x);
        break;
    }
}
```

C++:

```
cpp
CopyEdit
set<int> seen;
for (int i = 0; i < n; i++) {
    if (seen.find(arr[i]) != seen.end()) {
        cout << "Duplicate: " << arr[i];
        break;
    }
    seen.insert(arr[i]);
}
```

Python:

```
python
CopyEdit
seen = set()
for x in arr:
    if x in seen:
        print("Duplicate:", x)
        break
    seen.add(x)
```

✅ 10. Leaders in an Array

🧠 *Elements greater than all elements to their right.*

Java:

```
java
CopyEdit
int max = Integer.MIN_VALUE;
for (int i = arr.length - 1; i >= 0; i--) {
    if (arr[i] > max) {
        System.out.print(arr[i] + " ");
        max = arr[i];
    }
}
```

C++:

```
cpp
CopyEdit
int max = INT_MIN;
for (int i = n - 1; i >= 0; i--) {
    if (arr[i] > max) {
        cout << arr[i] << " ";
        max = arr[i];
    }
}
```

Python:

```
python
```

```
CopyEdit
max_so_far = float('-inf')
for x in reversed(arr):
    if x > max_so_far:
        print(x, end=' ')
        max_so_far = x
```

✅ 11. Maximum Subarray Sum (Kadane's Algo)

🧠 *Find the maximum possible sum of a contiguous subarray.*

Java:

```
java
CopyEdit
int maxSoFar = arr[0], curr = arr[0];
for (int i = 1; i < arr.length; i++) {
    curr = Math.max(arr[i], curr + arr[i]);
    maxSoFar = Math.max(maxSoFar, curr);
}
```

C++:

```
cpp
CopyEdit
int maxSoFar = arr[0], curr = arr[0];
for (int i = 1; i < n; i++) {
    curr = max(arr[i], curr + arr[i]);
    maxSoFar = max(maxSoFar, curr);
}
```

Python:

```
python
CopyEdit
max_so_far = curr = arr[0]
for x in arr[1:]:
    curr = max(x, curr + x)
    max_so_far = max(max_so_far, curr)
```

✅ 12. Stock Buy and Sell for Max Profit

🧠 *Buy low and sell high later.*

Java:

```
java
CopyEdit
int min = arr[0], profit = 0;
for (int i = 1; i < arr.length; i++) {
    profit = Math.max(profit, arr[i] - min);
    min = Math.min(min, arr[i]);
}
```

C++:

```
cpp
CopyEdit
int minVal = arr[0], profit = 0;
for (int i = 1; i < n; i++) {
    profit = max(profit, arr[i] - minVal);
    minVal = min(minVal, arr[i]);
}
```

Python:

```
python
CopyEdit
min_price = arr[0]
profit = 0
for price in arr[1:]:
    profit = max(profit, price - min_price)
    min_price = min(min_price, price)
```

✅ 13. Sort 0s, 1s, and 2s (Dutch National Flag)

🧠 *Sort an array of only 0s, 1s, and 2s in-place.*

Java:

```
java
CopyEdit
int low = 0, mid = 0, high = arr.length - 1;
while (mid <= high) {
    if (arr[mid] == 0) swap(arr, low++, mid++);
    else if (arr[mid] == 1) mid++;
    else swap(arr, mid, high--);
}
```

C++:

```
cpp
CopyEdit
int low = 0, mid = 0, high = n - 1;
while (mid <= high) {
    if (arr[mid] == 0) swap(arr[low++], arr[mid++]);
    else if (arr[mid] == 1) mid++;
    else swap(arr[mid], arr[high--]);
}
```

Python:

```
python
CopyEdit
low, mid, high = 0, 0, len(arr) - 1
while mid <= high:
    if arr[mid] == 0:
        arr[low], arr[mid] = arr[mid], arr[low]
        low += 1; mid += 1
```

```
    elif arr[mid] == 1:
        mid += 1
    else:
        arr[mid], arr[high] = arr[high], arr[mid]
        high -= 1
```

✅ 14. First Repeating Element

🧠 *Find the first element that repeats in array.*

Java:

```
java
CopyEdit
Map<Integer, Integer> map = new LinkedHashMap<>();
for (int val : arr) map.put(val, map.getOrDefault(val, 0) + 1);
for (int val : arr) {
    if (map.get(val) > 1) {
        System.out.println(val);
        break;
    }
}
```

C++:

```
cpp
CopyEdit
map<int, int> count;
for (int i = 0; i < n; i++) count[arr[i]]++;
for (int i = 0; i < n; i++) {
    if (count[arr[i]] > 1) {
        cout << arr[i]; break;
    }
}
```

Python:

```
python
CopyEdit
from collections import Counter
count = Counter(arr)
for x in arr:
    if count[x] > 1:
        print(x)
        break
```

✅ 15. Subarray with Given Sum (Positive Only)

🧠 *Find a subarray whose sum is equal to target.*

Java:

```
java
```

```
CopyEdit
int sum = 0, start = 0;
for (int end = 0; end < arr.length; end++) {
    sum += arr[end];
    while (sum > target) sum -= arr[start++];
    if (sum == target) {
        System.out.println("Found between " + start + " and " + end);
        break;
    }
}
```

C++:

```
cpp
CopyEdit
int sum = 0, start = 0;
for (int end = 0; end < n; end++) {
    sum += arr[end];
    while (sum > target) sum -= arr[start++];
    if (sum == target) {
        cout << "Found: " << start << " to " << end;
        break;
    }
}
```

Python:

```
python
CopyEdit
sum_, start = 0, 0
for end in range(len(arr)):
    sum_ += arr[end]
    while sum_ > target:
        sum_ -= arr[start]
        start += 1
    if sum_ == target:
        print("Found from", start, "to", end)
        break
```

✅ 15. Count Pairs with Given Sum

🧠 *Count the number of pairs whose sum equals a given target.*

Java:

```
java
CopyEdit
Map<Integer, Integer> map = new HashMap<>();
int count = 0;
for (int num : arr) {
    count += map.getOrDefault(target - num, 0);
    map.put(num, map.getOrDefault(num, 0) + 1);
}
```

C++:

```
cpp
```

```
CopyEdit
unordered_map<int, int> map;
int count = 0;
for (int num : arr) {
    count += map[target - num];
    map[num]++;
}
```

Python:

```
python
CopyEdit
from collections import defaultdict
count = 0
freq = defaultdict(int)
for num in arr:
    count += freq[target - num]
    freq[num] += 1
```

✅ 16. Longest Subarray with Sum = K

🧠 *Find the longest subarray that sums up to K.*

Java:

```
java
CopyEdit
Map<Integer, Integer> map = new HashMap<>();
int sum = 0, maxLen = 0;
for (int i = 0; i < arr.length; i++) {
    sum += arr[i];
    if (sum == k) maxLen = i + 1;
    if (!map.containsKey(sum)) map.put(sum, i);
    if (map.containsKey(sum - k))
        maxLen = Math.max(maxLen, i - map.get(sum - k));
}
```

C++:

```
cpp
CopyEdit
unordered_map<int, int> map;
int sum = 0, maxLen = 0;
for (int i = 0; i < n; i++) {
    sum += arr[i];
    if (sum == k) maxLen = i + 1;
    if (map.find(sum) == map.end()) map[sum] = i;
    if (map.find(sum - k) != map.end())
        maxLen = max(maxLen, i - map[sum - k]);
}
```

Python:

```
python
CopyEdit
prefix = {}
```

```
sum = 0
max_len = 0
for i in range(len(arr)):
    sum += arr[i]
    if sum == k:
        max_len = i + 1
    if sum - k in prefix:
        max_len = max(max_len, i - prefix[sum - k])
    if sum not in prefix:
        prefix[sum] = i
```

✅ 17. Rearrange Alternating Positive & Negative

🧠 *Rearrange so positives and negatives alternate, starting with positive.*

Java:

```
java
CopyEdit
List<Integer> pos = new ArrayList<>();
List<Integer> neg = new ArrayList<>();
for (int num : arr) {
    if (num >= 0) pos.add(num);
    else neg.add(num);
}
for (int i = 0; i < arr.length; i++) {
    if (i % 2 == 0 && !pos.isEmpty()) arr[i] = pos.remove(0);
    else if (!neg.isEmpty()) arr[i] = neg.remove(0);
}
```

C++:

```
cpp
CopyEdit
vector<int> pos, neg;
for (int num : arr) {
    if (num >= 0) pos.push_back(num);
    else neg.push_back(num);
}
int i = 0, p = 0, n = 0;
while (p < pos.size() && n < neg.size()) {
    arr[i++] = pos[p++];
    arr[i++] = neg[n++];
}
while (p < pos.size()) arr[i++] = pos[p++];
while (n < neg.size()) arr[i++] = neg[n++];
```

Python:

```
python
CopyEdit
pos = [x for x in arr if x >= 0]
neg = [x for x in arr if x < 0]
i = 0
while pos and neg:
    arr[i] = pos.pop(0)
```

```
    arr[i+1] = neg.pop(0)
    i += 2
arr[i:] = pos + neg
```

✅ 18. Trapping Rain Water

🧠 *Calculate water that can be trapped after raining.*

Java:

```
java
CopyEdit
int n = height.length;
int[] left = new int[n], right = new int[n];
left[0] = height[0];
right[n - 1] = height[n - 1];
for (int i = 1; i < n; i++)
    left[i] = Math.max(left[i - 1], height[i]);
for (int i = n - 2; i >= 0; i--)
    right[i] = Math.max(right[i + 1], height[i]);
int water = 0;
for (int i = 0; i < n; i++)
    water += Math.min(left[i], right[i]) - height[i];
```

C++:

```
cpp
CopyEdit
vector<int> left(n), right(n);
left[0] = height[0];
right[n-1] = height[n-1];
for (int i = 1; i < n; i++)
    left[i] = max(left[i-1], height[i]);
for (int i = n - 2; i >= 0; i--)
    right[i] = max(right[i+1], height[i]);
int water = 0;
for (int i = 0; i < n; i++)
    water += min(left[i], right[i]) - height[i];
```

Python:

```
python
CopyEdit
n = len(height)
left = [0]*n
right = [0]*n
left[0] = height[0]
right[-1] = height[-1]
for i in range(1, n):
    left[i] = max(left[i-1], height[i])
for i in range(n-2, -1, -1):
    right[i] = max(right[i+1], height[i])
water = sum(min(left[i], right[i]) - height[i] for i in range(n))
```

✅ 19. Stock Buy and Sell (Multiple Transactions)

🧠 *Buy and sell as many times as needed to maximize profit.*

Java:

```
java
CopyEdit
int profit = 0;
for (int i = 1; i < arr.length; i++) {
    if (arr[i] > arr[i - 1]) profit += arr[i] - arr[i - 1];
}
```

C++:

```
cpp
CopyEdit
int profit = 0;
for (int i = 1; i < n; i++) {
    if (arr[i] > arr[i-1]) profit += arr[i] - arr[i-1];
}
```

Python:

```
python
CopyEdit
profit = 0
for i in range(1, len(prices)):
    if prices[i] > prices[i - 1]:
        profit += prices[i] - prices[i - 1]
```

✅ 20. Subarray with Given Sum (Positive Integers)

🧠 *Find if a subarray with sum = target exists (for +ve only).*

Java:

```
java
CopyEdit
int sum = arr[0], start = 0;
for (int end = 1; end <= arr.length; end++) {
    while (sum > target && start < end - 1)
        sum -= arr[start++];
    if (sum == target)
        System.out.println("Found from " + start + " to " + (end - 1));
    if (end < arr.length)
        sum += arr[end];
}
```

C++:

```
cpp
CopyEdit
int sum = arr[0], start = 0;
```

```
for (int end = 1; end <= n; end++) {
    while (sum > target && start < end - 1)
        sum -= arr[start++];
    if (sum == target)
        cout << "Found from " << start << " to " << end - 1;
    if (end < n)
        sum += arr[end];
}
```

Python:

```
python
CopyEdit
sum = arr[0]
start = 0
for end in range(1, len(arr)+1):
    while sum > target and start < end - 1:
        sum -= arr[start]
        start += 1
    if sum == target:
        print(f"Found from {start} to {end-1}")
    if end < len(arr):
        sum += arr[end]
```