Array Problems

1. Sum of Array Elements
Write a Java program to calculate the sum of all elements in an integer array.
Example:

Input: arr[] = {1, 2, 3, 4, 5}
Output: 15

2. Find the Largest Element
Write a program to find the maximum element in an array.

Input: arr[] = {10, 25, 47, 85, 12}
Output: 85

3. Find the Smallest Element
Modify the previous program to find the smallest element in an array.

Input: arr[] = {10, 25, 47, 85, 12}
Output: 10

4. Reverse an Array
Write a Java program to reverse an array in place.

Input: arr[] = {1, 2, 3, 4, 5}
Output: {5, 4, 3, 2, 1}

5. Check if an Array is Sorted
Write a method to check whether an array is sorted in ascending order.

Input: arr[] = {1, 2, 3, 4, 5}
Output: true
Input: arr[] = {1, 3, 2, 4, 5}
Output: false

6. Count Even and Odd Numbers
Write a Java program to count the number of even and odd numbers in an array.

Input: arr[] = {1, 2, 3, 4, 5, 6}
Output: Even: 3, Odd: 3

7. Left Rotate an Array by One Position
Write a Java program to left rotate an array by one position.

Input: arr[] = {10, 20, 30, 40, 50}
Output: {20, 30, 40, 50, 10}

8. Find the Second Largest Number
Write a program to find the second largest element in an array without sorting.

Input: arr[] = {10, 20, 4, 45, 99}
Output: 45

9. Find Duplicate Elements
Write a Java program to find all duplicate elements in an array.

Input: arr[] = {4, 5, 6, 7, 5, 4, 9}
Output: 4, 5

10. Find Non-Repeating Elements
Write a Java program to print elements that appear only once in an array.

Input: arr[] = {1, 2, 3, 2, 1, 4}
Output: 3, 4

11. Merge Two Sorted Arrays
Write a Java program to merge two sorted arrays into a single sorted array.

Input: arr1[] = {1, 3, 5}, arr2[] = {2, 4, 6}
Output: {1, 2, 3, 4, 5, 6}

12. Move All Zeros to End
Write a program to move all 0s to the end of the array without changing the order of other elements.

Input: arr[] = {1, 0, 2, 0, 4, 3, 0}
Output: {1, 2, 4, 3, 0, 0, 0}

13. Find the Intersection of Two Arrays
Write a Java program to find the common elements in two integer arrays.

Input: arr1[] = {1, 2, 3, 4}, arr2[] = {3, 4, 5, 6}
Output: {3, 4}

14. Subarray with Given Sum
Write a program to find a continuous subarray that adds up to a given sum.

Input: arr[] = {1, 2, 3, 7, 5}, sum = 12
Output: {2, 3, 7}

15. Find the Missing Number in an Array
Given an array of size n-1 containing numbers from 1 to n, find the missing number.

Input: arr[] = {1, 2, 4, 5, 6}
Output: 3

16. Find Pairs with a Given Sum
Write a program to find pairs in an array that sum up to a given value.

Input: arr[] = {1, 5, 7, -1, 5}, sum = 6
Output: (1,5), (7,-1)

17. Find the Majority Element
An element in an array is a majority element if it appears more than n/2 times. Find the majority element (if any).

Input: arr[] = {3, 3, 4, 2, 4, 4, 2, 4, 4}
Output: 4

18. Rearrange Array in Alternating Positive & Negative Numbers
Write a program to rearrange an array such that positive and negative numbers appear alternately.

Input: arr[] = {1, -2, 3, -4, 5, -6}
Output: {-2, 1, -4, 3, -6, 5}

19. Find the Largest Sum Contiguous Subarray (Kadane's Algorithm)
Write a program to find the maximum sum of a contiguous subarray.

Input: arr[] = {-2, -3, 4, -1, -2, 1, 5, -3}
Output: 7

20. Spiral Order Traversal of a Matrix
Given a 2D matrix, print its elements in spiral order.

Input:
1 2 3
4 5 6
7 8 9
Output: {1, 2, 3, 6, 9, 8, 7, 4, 5}

21. Find the Next Greater Element
For each element in an array, find the next greater element (NGE) on its right. If there is no greater element, print -1.
Use Stack for optimal solution (O(n) complexity).

Input: arr[] = {4, 5, 2, 10, 8}
Output: {5, 10, 10, -1, -1}

22. Find the Smallest Window to be Sorted
Find the smallest subarray that must be sorted in order for the entire array to be sorted.

Input: arr[] = {1, 3, 5, 8, 7, 6, 9, 10}
Output: {5, 8, 7, 6} // Sorting this subarray will make the entire array sorted.

23. Find the Longest Consecutive Sequence
Given an unsorted array, find the length of the longest sequence of consecutive integers.
Use HashSet for an optimal O(n) solution.

Input: arr[] = {100, 4, 200, 1, 3, 2}
Output: 4 // The sequence {1, 2, 3, 4} has length 4.

24. Find the Maximum Product Subarray
Find the maximum product that can be obtained from a subarray.
Use Kadane's-like approach with two variables (max & min).

Input: arr[] = {2, 3, -2, 4}
Output: 6 // {2, 3} gives the maximum product.

25. Count Inversions in an Array (Merge Sort Approach)
An inversion in an array is when a larger number appears before a smaller one. Count the number of inversions.
Use Merge Sort for an O(n log n) solution.

Input: arr[] = {8, 4, 2, 1}
Output: 6 // Pairs: (8,4), (8,2), (8,1), (4,2), (4,1), (2,1)

26. Find the Peak Element
A peak element is an element greater than its neighbors. Find a peak element's index.
Use Binary Search for O(log n) complexity.

Input: arr[] = {1, 3, 20, 4, 1, 0}
Output: 2 // arr[2] = 20 is a peak.

27. Trapping Rainwater Problem
Given an array where each index represents height, find how much water is trapped after rain.
Use two-pointer approach for an O(n) solution.

Input: arr[] = {0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1}
Output: 6

28. Find All Triplets with Zero Sum
Find all unique triplets in the array whose sum is 0.
Use Sorting + Two-Pointer technique for $O(n^2)$.

Input: arr[] = {-1, 0, 1, 2, -1, -4}
Output: (-1, 0, 1), (-1, -1, 2)

29. Rearrange Array in Zig-Zag Order
Rearrange an array such that a < b > c < d > e...

Input: arr[] = {4, 3, 7, 8, 6, 2, 1}
Output: {3, 7, 4, 8, 2, 6, 1}

30. Maximum Sum of i * arr[i] Rotations
Find the maximum sum of i * arr[i] by rotating the array.
Use a mathematical approach to optimize in O(n).

Input: arr[] = {1, 2, 3, 4}
Output: 20 // Rotation {4, 1, 2, 3} gives max sum.

31. Find the First Missing Positive Integer
Find the smallest missing positive number in an array of size n.
Use index mapping for O(n) complexity.

Input: arr[] = {3, 4, -1, 1}
Output: 2

32. Find Pythagorean Triplets in an Array
Check if there exist three numbers that satisfy $a^2 + b^2 = c^2$.

Input: arr[] = {3, 1, 4, 6, 5}
Output: true // (3, 4, 5) forms a triplet.

33. Find the Celebrity in a Party
A celebrity is a person who knows no one but is known by everyone. Find the celebrity in an N × N matrix.

Use a stack-based O(n) solution.

Input:
[[0, 1, 1],
  [0, 0, 0],
  [1, 1, 0]]
Output: Person at index 1 is the celebrity.

34. Maximum Difference Between Two Elements
Find the maximum difference between two elements such that the larger number appears after the smaller one.

Input: arr[] = {2, 3, 10, 6, 4, 8, 1}
Output: 8 // 10 - 2 = 8

35. Find the Largest Rectangle in Histogram
Given an array representing bar heights, find the largest rectangular area possible in a histogram.
Use Stack for an O(n) solution.

Input: heights[] = {6, 2, 5, 4, 5, 1, 6}
Output: 12

36. Find Maximum Length of Bitonic Subarray
A bitonic subarray is a subarray that first increases and then decreases. Find the length of the longest bitonic subarray.

Input: arr[] = {1, 2, 5, 3, 2, 8, 9, 10, 7, 6}
Output: 7 // {2, 5, 3, 2, 8, 9, 10}

37. Count Number of Subarrays with XOR = K
Given an array of integers and an integer K, count the number of subarrays whose XOR is equal to K.
Use HashMap for an O(n) solution.

Input: arr[] = {4, 2, 2, 6, 4}, K = 6
Output: 4

38. Find Maximum Subarray Sum After One Deletion
Find the maximum sum of a subarray where you are allowed to delete one element.
Use Kadane's algorithm with a tracking variable.

Input: arr[] = {1, -2, 0, 3}
Output: 4 // {1, 3}

39. Find Median of Two Sorted Arrays
Given two sorted arrays, find their median in O(log n) time.
Use Binary Search.

Input: arr1[] = {1, 3, 8}, arr2[] = {7, 9, 10, 11}
Output: 8

40. Find Maximum Distance Between Occurrences of Same Element
Find the maximum index difference j - i such that arr[i] == arr[j].

Input: arr[] = {1, 2, 3, 4, 1, 5, 1}
Output: 6 // Index difference (6 - 0)

41. Smallest Subarray with Sum Greater than X
Find the smallest subarray whose sum is greater than X.
Use Sliding Window for O(n) complexity.

Input: arr[] = {1, 4, 45, 6, 10, 19}, X = 51
Output: 3 // {4, 45, 6}

42. Longest Increasing Subsequence (LIS)
Find the length of the longest increasing subsequence in an array.
Use Dynamic Programming for $O(n^2)$ or Binary Search for O(n log n).

Input: arr[] = {10, 22, 9, 33, 21, 50, 41, 60, 80}
Output: 6 // {10, 22, 33, 50, 60, 80}

43. Merge Overlapping Intervals
Given a list of intervals, merge all overlapping intervals.

Input: intervals = {{1,3}, {2,6}, {8,10}, {15,18}}
Output: {{1,6}, {8,10}, {15,18}}

44. Find Minimum Swaps to Sort an Array
Find the minimum number of swaps required to sort an array.

Input: arr[] = {4, 3, 2, 1}
Output: 2 // (4↔1, 3↔2)

45. Find the Missing and Repeating Number
Given an array of size n where numbers are from 1 to n, find the missing and repeating numbers.
Use XOR for an O(n) solution.

Input: arr[] = {4, 3, 6, 2, 1, 1}
Output: Missing: 5, Repeating: 1

46. Maximum Absolute Difference
Find the maximum absolute difference between arr[i] - arr[j] + i - j.

Input: arr[] = {1, 3, -1}
Output: 5

47. Count All Possible Triangles
Given an array of side lengths, count the number of triangles that can be formed.

Input: arr[] = {4, 6, 3, 7}
Output: 3

48. Find Longest Palindromic Subarray
Find the longest subarray that is a palindrome.

Input: arr[] = {1, 2, 3, 2, 1, 5, 6}
Output: {1, 2, 3, 2, 1}

49. Find the Maximum Length of a Subarray with Sum 0
Find the longest contiguous subarray with sum 0.
Use HashMap for O(n).

Input: arr[] = {1, 2, -3, 3, 1}
Output: 3

50. Maximum Profit with K Transactions
Find the maximum profit possible with at most K stock transactions.
Use Dynamic Programming for O(nK) complexity.

Input: prices[] = {3,2,6,5,0,3}, K = 2
Output: 7

51. Find Minimum Moves to Make Array Elements Equal
Find the minimum number of moves required to make all elements equal.
Use median-based approach.

Input: arr[] = {1, 2, 3}
Output: 2 // Change 1 to 2, and 3 to 2

52. Find Maximum Area of Rectangle in a Binary Matrix

Given a binary matrix, find the largest rectangle with all 1s.
Use Histogram-based Stack approach.

Input: matrix =
1 0 1 0 0
1 0 1 1 1
1 1 1 1 1
1 0 0 1 0
Output: 6

53. Find Pairs with Minimum Absolute Difference
Find pairs of numbers with the minimum absolute difference.

Input: arr[] = {3, 8, 15, 17}
Output: {(15, 17), (8, 3)}

54. Find the Largest Subarray with Equal Number of 0s and 1s
Find the longest subarray where the number of 0s and 1s are equal.
Use HashMap for O(n).

Input: arr[] = {0, 0, 1, 0, 1, 1, 1, 0}
Output: 6

55. Find the Smallest Subset with Sum Greater than Half of Total Sum
Find the smallest subset whose sum is greater than half of the total sum.

Input: arr[] = {3, 1, 7, 1}
Output: {7, 3}