JavaScript — 50 Detailed Array Coding Questions

Q1. Merge two arrays (concatenate without removing duplicates).

Given: arr1 = [1, 2, 4], arr2 = [4, 6, 7]
Goal: Return [1, 2, 4, 4, 6, 7]

Q2. Find common numbers (intersection, unique) between two arrays.

Given: arr1 = [3, 5, 6], arr2 = [4, 5, 7]
Goal: Return [5]

Q3. Reverse the array *without using* any inbuilt array methods (no reverse, no push/pop). Use two-pointer swapping or manual loop.

Given: arr = [6, 7, 9, 1, 5]
Goal: Return [5, 1, 9, 7, 6]

Q4. Remove duplicate values from an array while preserving the first occurrence order (no Set).

Given: arr = [2, 3, 2, 4, 3, 5]
Goal: Return [2, 3, 4, 5]

Q5. Rotate the array to the left by k steps (in-place if possible).

Given: arr = [1, 2, 3, 4, 5], k = 2
Goal: Return [3, 4, 5, 1, 2]

Q6. Rotate the array to the right by k steps (in-place if possible).

Given: arr = [10, 20, 30, 40, 50], k = 1
Goal: Return [50, 10, 20, 30, 40]

Q7. Compute the sum of all numbers in the array (no reduce).

Given: arr = [5, 8, 2, 4]
Goal: Return 19

Q8. Compute the product of all numbers in the array.

Given: arr = [1, 3, 2, 4]
Goal: Return 24

Q9. Find the maximum number in the array (no Math.max spread).

Given: arr = [9, 2, 17, 6, 4]
Goal: Return 17

Q10. Find the minimum number in the array.

Given: arr = [9, 2, 17, 6, 4]
Goal: Return 2

Q11. Find the second largest number in the array (distinct values).

Given: arr = [12, 7, 9, 15, 12]

Goal: Return 12

Q12. Count occurrences of a given number in the array.

Given: arr = [3, 5, 3, 2, 3, 1], target = 3
Goal: Return 3

Q13. Find the index of the first occurrence of a target value (manual search).

Given: arr = [4, 7, 9, 7, 2], target = 7
Goal: Return 1

Q14. Find the index of the last occurrence of a target value (manual search).

Given: arr = [4, 7, 9, 7, 2], target = 7
Goal: Return 3

Q15. Check if the array is sorted in non-decreasing order.

Given: arr = [1, 2, 2, 4, 5]
Goal: Return true

Q16. Sort an array of small size using Bubble Sort (no sort()).

Given: arr = [5, 1, 4, 2, 8]
Goal: Return [1, 2, 4, 5, 8]

Q17. Merge two *sorted* arrays into a single sorted array (two pointers).

Given: arr1 = [1, 3, 5], arr2 = [2, 4, 6]
Goal: Return [1, 2, 3, 4, 5, 6]

Q18. Remove all falsy values (false, 0, '', NaN, null, undefined) from an array without using filter().

Given: arr = [0, 1, false, 2, '', 3, null]
Goal: Return [1, 2, 3]

Q19. Insert a value at a given index (shift elements right, no splice).

Given: arr = [10, 20, 30, 40], index = 2, value = 99
Goal: Return [10, 20, 99, 30, 40]

Q20. Delete the element at a given index (shift elements left, no splice).

Given: arr = [10, 20, 30, 40], index = 1
Goal: Return [10, 30, 40]

Q21. Split an array into chunks of size n (last chunk may be smaller).

Given: arr = [1, 2, 3, 4, 5], n = 2
Goal: Return [[1, 2], [3, 4], [5]]

Q22. Flatten a one-level nested array without using flat().

Given: arr = [1, [2, 3], 4, [5]]
Goal: Return [1, 2, 3, 4, 5]

Q23. Elements in arr1 that are not in arr2 (relative difference).

Given: arr1 = [1, 2, 3, 4], arr2 = [3, 4, 5]
Goal: Return [1, 2]

Q24. Union of two arrays (unique values, order by first appearance).

Given: arr1 = [1, 2, 2], arr2 = [2, 3]
Goal: Return [1, 2, 3]

Q25. Intersection with counts (multiset intersection).

Given: arr1 = [1, 2, 2, 3], arr2 = [2, 2, 4]
Goal: Return [2, 2]

Q26. Build a frequency map (object) counting how many times each element appears.

Given: arr = ['a', 'b', 'a', 'c', 'b', 'a']
Goal: Return {'a': 3, 'b': 2, 'c': 1}

Q27. Find the most frequent element in an array; if tie, return any one.

Given: arr = [2, 3, 2, 4, 4, 4, 3]
Goal: Return 4

Q28. Remove all occurrences of a value from an array (no filter).

Given: arr = [3, 1, 3, 2, 3, 4], value = 3
Goal: Return [1, 2, 4]

Q29. Replace all occurrences of a value with another value.

Given: arr = [1, 2, 2, 3, 2], from = 2, to = 9
Goal: Return [1, 9, 9, 3, 9]

Q30. Compute element-wise sum of two arrays of equal length.

Given: arr1 = [1, 2, 3], arr2 = [4, 5, 6]
Goal: Return [5, 7, 9]

Q31. Compute prefix sums (cumulative sums).

Given: arr = [2, 1, 3, 4]
Goal: Return [2, 3, 6, 10]

Q32. Reverse an array in-place using two pointers (demonstrate indices swapping).

Given: arr = [11, 22, 33, 44]
Goal: Return [44, 33, 22, 11]

Q33. Find any two numbers that sum to a target (return the pair or indices).

Given: arr = [1, 4, 5, 3, 7], target = 8
Goal: Return [1, 7] or indices [0, 4]

Q34. Find all pairs of numbers that sum to a target (no duplicates, i < j).

Given: arr = [2, 4, 3, 5, 7, 8, 9], target = 7
Goal: Return [[2, 5], [4, 3]] (order may vary)

Q35. Move all zeros to the end while maintaining the relative order of non-zero elements (in-place).

Given: arr = [0, 1, 0, 3, 12]
Goal: Return [1, 3, 12, 0, 0]

Q36. Partition array into evens first then odds, preserving original order within each group.

Given: arr = [3, 8, 5, 12, 7, 6]
Goal: Return [8, 12, 6, 3, 5, 7]

Q37. Compute the average (mean) of numbers in the array (integer or float).

Given: arr = [10, 20, 30]
Goal: Return 20

Q38. Find the median of an odd-length array (you may sort or use selection).

Given: arr = [7, 1, 3]
Goal: Return 3

Q39. Given a sorted array, remove adjacent duplicates in-place and return the new length; also return the resulting array prefix.

Given: arr = [1, 1, 2, 2, 3]
Goal: New length 3, array becomes [1, 2, 3, _, _]

Q40. Find the length of the longest strictly increasing contiguous subarray.

Given: arr = [1, 2, 2, 3, 4, 1]
Goal: Return 3 (for [2, 3, 4])

Q41. Create an array of the first n squares.

Given: n = 5
Goal: Return [1, 4, 9, 16, 25]

Q42. Create an array with numbers from 1 to N (no Array.from).

Given: N = 6
Goal: Return [1, 2, 3, 4, 5, 6]

Q43. Reverse an array of strings (e.g., words) and join with spaces.

Given: arr = ['I', 'love', 'JS']
Goal: Return 'JS love I'

Q44. Convert an array of numbers to strings (no map).

Given: arr = [1, 2, 3]
Goal: Return ['1', '2', '3']

Q45. Check if every element is even (no every()).

Given: arr = [2, 4, 6, 8]
Goal: Return true

Q46. Check if at least one element is greater than a threshold (no some()).

Given: arr = [4, 9, 1], threshold = 8
Goal: Return true

Q47. Find the index of the smallest element (first for ties).

Given: arr = [9, 4, 7, 2, 5]
Goal: Return 3

Q48. Swap the positions of the minimum and maximum elements in the array.

Given: arr = [5, 1, 9, 3]
Goal: Return [5, 9, 1, 3]

Q49. Shuffle the array using the Fisher–Yates algorithm (show final shuffled array; randomness may vary).

Given: arr = [1, 2, 3, 4]
Goal: Return a permutation like [3, 1, 4, 2]

Q50. Remove duplicates but keep first occurrence order (stable unique, no Set).

Given: arr = [4, 5, 4, 4, 2, 5, 1]
Goal: Return [4, 5, 2, 1]