100 JavaScript Programming Problems (Beginner to Advanced)

Level 1: Very Basic

1. Print 'Hello, World!' to the console.
2. Declare a variable and print its value.
3. Take two numbers and print their sum.
4. Convert a string to a number.
5. Check whether a number is even or odd.
6. Find the square of a number.
7. Swap two variables (with and without a third variable).
8. Find the largest of two numbers.
9. Find the largest of three numbers.
10. Convert Celsius to Fahrenheit.
11. Convert minutes into hours and minutes.
12. Check if a number is positive, negative, or zero.
13. Print numbers from 1 to 10.
14. Print all even numbers from 1 to 50.
15. Print the multiplication table of a number.

Level 2: Conditions & Loops

1. Calculate the factorial of a number.
2. Print the sum of numbers from 1 to N.
3. Reverse a number.
4. Check if a number is a palindrome.
5. Count the number of digits in a number.
6. Find the power of a number without using Math.pow.
7. Print Fibonacci series up to N terms.
8. Check if a number is prime.
9. Print all prime numbers between 1 and 100.
10. Find the LCM of two numbers.
11. Find the GCD/HCF of two numbers.
12. Print a right triangle star pattern.
13. Print a pyramid star pattern.
14. Print a number pattern.
15. Sum of even and odd digits in a number.

Level 3: Functions

1. Write a function to add two numbers.
2. Write a function to check if a number is prime.
3. Write a function to return the maximum element in an array.
4. Write a function to reverse a string.
5. Write a function to check if a string is a palindrome.
6. Write a function to count vowels in a string.
7. Write a function to capitalize the first letter of each word.
8. Write a function to find factorial using recursion.
9. Write a function to generate Fibonacci using recursion.
10. Write a function that returns the sum of array elements.
11. Write a function to remove duplicates from an array.
12. Write a function to merge two arrays.
13. Write a function to find the average of numbers.
14. Write a function to check anagrams.
15. Write a function that returns a random number in a given range.

Level 4: Arrays

1. Find the largest and smallest number in an array.
2. Sort an array ascending and descending.
3. Find the second largest number in an array.
4. Count even and odd numbers in an array.
5. Remove falsy values from an array.
6. Rotate an array left by one position.
7. Rotate an array right by K positions.
8. Find the missing number in an array from 1 to N.
9. Find duplicate elements in an array.
10. Find the intersection of two arrays.
11. Find the union of two arrays.
12. Flatten a nested array.
13. Chunk an array into smaller arrays.
14. Find the most frequent element in an array.
15. Move all zeros to the end of an array.

Level 5: Strings

1. Reverse a string without using reverse().
2. Count the frequency of characters in a string.
3. Find the first non-repeating character.
4. Remove all spaces from a string.
5. Replace all vowels with '*'.
6. Check if two strings are rotations of each other.
7. Find the longest word in a sentence.
8. Convert a string to camelCase.
9. Convert a string to snake_case.
10. Check if a string contains only digits.

Level 6: Objects

1. Create an object and print all keys and values.
2. Count the number of properties in an object.
3. Merge two objects.
4. Find the property with the highest value.
5. Convert an object into an array of key-value pairs.
6. Group array of objects by a property.
7. Sort an array of objects by a key.
8. Remove a property from an object immutably.
9. Deep clone an object.
10. Compare two objects for deep equality.

Level 7: Advanced Logic

1. Implement linear search.
2. Implement binary search.
3. Find the longest substring without repeating characters.
4. Check if parentheses are balanced.
5. Find the maximum subarray sum (Kadane’s Algorithm).
6. Implement debounce.
7. Implement throttle.
8. Implement a custom map() function.
9. Implement a custom filter() function.
10. Implement a custom reduce() function.

Level 8: Async & Real World

1. Create a Promise that resolves after 2 seconds.
2. Handle Promise rejection.
3. Fetch data from an API and display it.
4. Convert callback-based code to Promises.
5. Use async/await with error handling.
6. Implement a retry mechanism for failed API calls.
7. Execute multiple Promises in parallel.
8. Implement a simple event emitter.
9. Build a todo list logic (no UI).
10. Build a shopping cart logic (add, remove, total price).