Array-Practice-JS

Basic Level: Array Creation and Access

1. **Array Basics**: Create an array of your favorite fruits and print each fruit.
2. **Array Indexing**: Given an array of numbers `[1, 2, 3, 4, 5]`, access and print the third element.
3. **Updating Elements**: Create an array of colors, and change the second color to “blue”.
4. **Length of Array**: Find and print the length of an array `[10, 20, 30, 40, 50]`.

Intermediate Level: Array Methods

1. **Adding and Removing Elements**: Create an array of animals. Add a new animal to the end, remove the first animal, and print the updated array.
2. **Concatenation**: Concatenate two arrays of colors and print the resulting array.
3. **Looping Through Array**: Create an array of numbers from 1 to 5. Loop through the array and print each number multiplied by 2.
4. **Array Includes**: Check if a certain element (e.g., 3) exists in an array `[1, 2, 3, 4, 5]`.
5. **Array Slice**: Given an array `[1, 2, 3, 4, 5]`, create a new array containing only the last three elements.
6. **Array Splice**: Remove the third element from an array of names and print the updated array.

Advanced Level: Array Problem Solving

1. **Finding Maximum Value**: Write a function to find the maximum value in an array of numbers.

2. **Finding Minimum Value**: Write a function to find the minimum value in an array of numbers.
3. **Sum of Array Elements**: Write a function to calculate the sum of all elements in an array.
4. **Average of Array Elements**: Write a function to calculate the average of elements in an array.
5. **Remove Duplicates**: Write a function to remove duplicate values from an array.
6. **Array Reversal**: Write a function to reverse the elements of an array without using the built-in `reverse` method.
7. **Merge and Sort Arrays**: Given two unsorted arrays, merge them into one sorted array.
8. **Find Intersection of Arrays**: Write a function to find the common elements between two arrays.
9. **Frequency Count**: Write a function to count the frequency of each element in an array.
10. **Unique Elements**: Given an array, return only the unique elements in a new array.

Expert Level: Complex Array Manipulations

1. **Moving Zeros**: Write a function to move all zeros to the end of an array, while maintaining the order of other elements.
2. **Rotate Array**: Write a function to rotate an array to the right by a specified number of steps.
3. **Array Chunking**: Write a function that divides an array into chunks of a specified size.
4. **Flatten Nested Arrays**: Write a function to flatten a nested array (e.g., `[[1, 2], [3, 4, [5, 6]]]`).
5. **Longest Consecutive Sequence**: Given an unsorted array of numbers, find the length of the longest consecutive sequence.

6. **Find Pairs with Sum**: Given an array and a target sum, find all pairs of numbers that add up to the target sum.
7. **Sort Array by Frequency**: Write a function that sorts an array by the frequency of its elements, with higher frequency elements coming first.
8. **Generate Subarrays**: Write a function that generates all possible subarrays of a given array.
9. **Partition Array**: Write a function to partition an array based on a given pivot element.
10. **Product of All Except Self**: Write a function to return an array where each element is the product of all elements except itself, without using division.
