Assignment 1 (JavaScript Functions)

- **Anagram Check**: Write a function isAnagram to check if two strings are anagrams.
- **Category Spending Calculation**: Write calculateTotalSpentByCategory to calculate total expenses by category from a list of transactions.
- **Find Largest Element**: Write findLargestElement to return the largest number in an array.
- **Palindrome Check**: Write isPalindrome to check if a string is a palindrome.
- **Execution Time Calculation**: Write a function to calculate the time taken to sum numbers from 1 to n.
- **Count Vowels**: Write countVowels to return the number of vowels in a string.
- **Array Sum**: Write sumArray to calculate the sum of numbers in an array.
- **Filter Even Numbers**: Write filterEvenNumbers to return only even numbers from an array.
- **Find Smallest Element**: Write findSmallestElement to return the smallest number in an array.
- **String Reversal**: Write reverseString to reverse a string.
- **Fibonacci Sequence**: Write fibonacci to get the nth Fibonacci number.
- **Remove Duplicates**: Write removeDuplicates to remove duplicates from an array.
- **Character Occurrence Count**: Write countOccurrences to count occurrences of a specific character in a string.
- **Find Common Elements**: Write findCommonElements to find common elements between two arrays.
- **String Sorting**: Write sortStrings to sort an array of strings alphabetically.

Assignment 2 (Solidity Smart Contracts)

- **Hello World Contract**: Create a contract to return “Hello, World!”.
- **Simple Storage Contract**: Create a contract to store and retrieve an integer.
- **Personalized Greeting Contract**: Create a contract to set and retrieve personalized greetings.
- **Counter Contract**: Create a contract to increment and retrieve a count.
- **Name Storage Contract**: Create a contract to store and retrieve a user’s name.
- **Basic Voting Contract**: Create a voting contract allowing users to vote and track votes.
- **Owner Access Control**: Create a contract with restricted functions accessible only by the owner.
- **Event Logging Contract**: Create a contract to log deposit and withdrawal events.
- **Transaction Ledger**: Create a contract to record and retrieve transactions.
- **Message Storage Contract**: Create a contract to store and retrieve messages.

Assignment 3 (Solidity Lottery System)

- **Lottery Contract**: Create a lottery contract where users pay a fee to enter, a winner is picked at random, and the manager sets an end time.

Assignment 4 (JavaScript and Solidity)

- **SHA-256 Hash Finder**: Write a function to find a SHA-256 hash starting with “00000”.
- **Hash with Custom Prefix**: Modify the hash finder to start with a given prefix.
- **Nonce Finder for Transaction Hash**: Write code to find a nonce for a specific transaction to get a hash starting with “00000”.
- **CryptoZombies Lesson**: Complete Lesson 1 on CryptoZombies.
- **Donation Contract**: Write a smart contract allowing donations to specified recipients.

Assignment 5 (Advanced Solidity Contracts)

- **Crowdfunding Contract**: Create a contract for crowdfunding, allowing users to create campaigns, contribute funds, and finalize campaigns with conditions for refunds or fund release.
- **Voting System Contract**: Create a voting contract for multiple proposals, allowing users to propose options, cast votes, and determine the winning proposal based on votes.

These assignments provide comprehensive practice in JavaScript programming and Solidity smart contract development, covering essential blockchain concepts and coding challenges.