String-Practice-JS

Basic Questions

1. **Creating Strings**: Create a string variable using single quotes, double quotes, and backticks (template literals).
2. **Concatenation**: Write a function `concatStrings(str1, str2)` that takes two strings and returns their concatenation.
3. **String Length**: Given a string `str`, find and print its length.
4. **Character Access**: Write a function `getFirstAndLastChar(str)` that returns the first and last characters of a string.
5. **Convert Case**: Create a function `toUpperAndLower(str)` that converts a string to uppercase and lowercase.

Intermediate Questions

1. **Template Literals**: Given a `name` variable, use a template literal to create a greeting: `"Hello, [name]!"`.
2. **IndexOf and LastIndexOf**: Write a function `findFirstAndLast(str, char)` that returns the first and last positions of a character in a string.
3. **Substring**: Write a function `getSubstring(str, start, end)` that extracts a substring from `str` between the start and end indices.
4. **Replace**: Write a function `replaceWord(sentence, target, replacement)` that replaces the first occurrence of `target` in `sentence` with `replacement`.
5. **String Split**: Write a function `splitSentence(sentence)` that splits a sentence into an array of words.

Advanced Questions

1. **Reverse String**: Write a function `reverseString(str)` that takes a string and returns it in reverse order.
2. **Count Words**: Create a function `wordCount(sentence)` that counts the number of words in a sentence.
3. **Palindrome Check**: Write a function `isPalindrome(str)` that checks if a given string is a palindrome (reads the same backward and forward).
4. **Frequency of Characters**: Write a function `charFrequency(str)` that returns an object with each character in the string as keys and their frequency as values.
5. **Truncate String**: Write a function `truncateString(str, num)` that truncates a string if it's longer than `num` characters and adds `"..."` at the end.

Challenging Questions

1. **Vowel Count**: Write a function `countVowels(str)` that counts the number of vowels in a string.
2. **Longest Word**: Create a function `findLongestWord(sentence)` that finds and returns the longest word in a sentence.
3. **Title Case Conversion**: Write a function `toTitleCase(sentence)` that converts each word in a sentence to start with an uppercase letter and the rest in lowercase.
4. **Remove Duplicate Characters**: Write a function `removeDuplicates(str)` that removes duplicate characters from a string.
5. **Anagram Check**: Write a function `isAnagram(str1, str2)` that checks if two strings are anagrams of each other (contain the same characters in different order).

Regex-Based Questions

1. **Extract Digits**: Write a function `extractDigits(str)` that extracts all numbers from a string and returns them as an array.
2. **Validate Email**: Write a function `isValidEmail(email)` that checks if a given string is a valid email address format.

3. **Remove Vowels**: Create a function `removeVowels(str)` that removes all vowels from a string.
4. **Count Specific Word**: Write a function `countWordOccurrences(sentence, word)` that counts the occurrences of a specific word in a sentence.
5. **Capitalize First Letter of Each Sentence**: Given a paragraph, write a function `capitalizeSentences(paragraph)` that capitalizes the first letter of each sentence.

String Manipulation Questions

1. **Caesar Cipher**: Write a function `caesarCipher(str, shift)` that encrypts a string by shifting each letter by a given number (Caesar cipher).
2. **Remove Whitespace**: Write a function `removeExtraSpaces(str)` that removes extra whitespace between words in a string.
3. **Mask Sensitive Information**: Write a function `maskString(str, visibleCount)` that replaces characters in `str` with `#` except for the last `visibleCount` characters.
4. **Find Common Prefix**: Create a function `findCommonPrefix(arr)` that finds the longest common prefix in an array of strings.
5. **Sort Words in Sentence**: Write a function `sortWords(sentence)` that sorts the words in a sentence alphabetically.
