

Question 1:

Write a function that takes an array of numbers as input and returns the sum of all the positive numbers in the array.

Question 2:

Implement a function that reverses a string without using the built-in `reverse()` function.

Question 3:

Write a function that takes a string as input and determines whether it is a palindrome (reads the same forwards and backwards). Ignore spaces and punctuation.

Question 4:

Given an array of integers, write a function that finds the two numbers that add up to a given target number. Return the indices of the two numbers in the array.

Question 5:

Implement a function to flatten a nested array. The function should take an array that may contain other arrays of integers and return a single array with all the integers.

Question 6:

**Write a function that takes two arrays of strings as input and returns a new array containing the common elements from both arrays, without any duplicates.**

```
function findCommonElements(arr1, arr2) {  
  
    const set1 = new Set(arr1);  
  
    const set2 = new Set(arr2);  
  
  
  
    const commonElements = [];
```

```
for (const element of set1) {  
  if (set2.has(element)) {  
    commonElements.push(element);  
  }  
}  
  
return commonElements;  
}  
  
const arr1 = ["apple", "banana", "orange", "grape"];  
const arr2 = ["orange", "kiwi", "banana", "pineapple"];  
const common = findCommonElements(arr1, arr2);  
console.log(common); // Output: ["banana", "orange"]
```

Question 7:

Implement a function that calculates the factorial of a given number using recursion.

Question 8:

**Write a function that takes a sorted array of numbers and a target number. The function should return the index of the target number in the array, or -1 if it is not found. Use a binary search algorithm for an efficient solution.**

Question 9:

**Given an array of integers, write a function that moves all the zeros to the end of the array while maintaining the order of the other elements.**

Question 10:

**Implement a function that takes a string as input and returns the most frequent character(s) in the string. If there is a tie, return all the characters that are tied.**

Question 1:

Write a function that takes an array of numbers as input and returns an object that contains the count of each number in the array.

For example, given the input: [1, 2, 3, 2, 1, 3, 3, 4, 5, 4], the function should return: {1: 2, 2: 2, 3: 3, 4: 2, 5: 1}.

Question 2:

**Write a function that takes an array of integers as input and returns the number that appears the most frequently in the array. If there is a tie, return any of the tied numbers.**

For example, given the input: [1, 2, 3, 2, 1, 3, 3, 4, 4, 5, 4], the function should return either 3 or 4.

Question 3:

Given an array of integers, write a function that finds all the duplicate numbers and returns them in an array.

For example, given the input: [1, 2, 3, 2, 1, 3, 3, 4, 4, 5, 4], the function should return: [1, 2, 3, 4].

Question 4:

**Write a function that takes an array of strings as input and returns a map where the keys are the strings and the values are the number of times each string appears in the array.**

For example, given the input: ["apple", "banana", "orange", "apple", "banana"], the function should return a map: { "apple": 2, "banana": 2, "orange": 1 }.

Question 5:

**Write a function that takes a map as input and returns an array of the keys sorted in descending order based on their values.**

For example, given the input: { "apple": 2, "banana": 4, "orange": 1 }, the function should return: ["banana", "apple", "orange"].