Important: In this exercise try to use the javascript programming language.

Easy

Exercice 1

Concatenating Two Words Entered by the User
This program prompts the user to input two separate words, concatenates them, and then displays the result in a prompt. The goal is to demonstrate how user input can be captured and manipulated to form a new string by joining two words together.
Requirements:
- The program should ask the user to enter two words, one at a time.
- It will then concatenate the two words without adding any spaces or additional characters between them.
- Finally, the program will display the concatenated string in a pop-up alert.

Example:
If the user inputs the words:
- First word: `Good`
- Second word: `Morning`
The output will be:
The concatenated result is: GoodMorning

Steps:
1. The user is prompted to enter the first word.
2. The user is then prompted to enter the second word.
3. Both words are concatenated (joined together as a single string).
4. The concatenated result is displayed in an alert.

Middle-Medium

Exercice 2: Count the letters in a string
Ask the user to enter a string, then display the number of letters it contains (ignoring spaces).

Exercise 3: sum of an array
Write a function that takes an array of integers as input and returns the sum of all its elements.

Exercice 4
Manipulate Array by Moving Zeroes to the Beginning

Exercice 5: Reverse a string
Ask the user to enter a string, then display the string reversed.

Medium

Exercice 4
Manipulate Array by Moving Zeroes to the Beginning

Write a JavaScript function that accepts an array of integers as input. The goal of this exercise is to traverse through the array, find any occurrence of the number `0`, and move all zeroes to the beginning of the array, while maintaining the relative order of the non-zero elements.

Requirements:

- The function should take an array of integers as input.
- If a `0` is found in the array, it should be removed from its current position and placed at the beginning of the array.
- The relative order of non-zero elements should not change.
- The function should return the modified array.

Example

Input: [1, 0, 3, 0, 5]
Output: [0, 0, 1, 3, 5]

Input: [0, 2, 0, 4, 0, 1]
Output: [0, 0, 0, 2, 4, 1]

Input: [7, 8, 9]
Output: [7, 8, 9]

Key Points to Consider:
- The function should handle edge cases, such as an array with no zeroes or an array that contains only zeroes.
- Aim for an efficient solution that minimizes the number of operations performed on the array.