Advanced Iteration Logic

Ex6.1 - Mumbling

This time no story, no theory. The examples below show you how to write function `accum`:
Examples:
accum("abcd") -> "A-Bb-Ccc-Dddd"
accum("RqaEzty") -> "R-Qq-Aaa-Eeeezzzz-Ttttt-Yyyyyyy"
accum("cwAt") -> "C-Ww-Aaa-Tttt"
The parameter of accum is a string which includes only letters from `a..z` and `A..Z`.

Ex6.2 - Counting Duplicates

Count the number of Duplicates
Write a function that will return the count of distinct case-insensitive alphabetic characters and numeric digits that occur more than once in the input string. The input string can be assumed to contain only alphabets (both uppercase and lowercase) and numeric digits.
Example
"abcde" -> 0 # no characters repeats more than once
"aabbcde" -> 2 # 'a' and 'b'
"aabBcde" -> 2 # 'a' occurs twice and 'b' twice (`b` and `B`)
"indivisibility" -> 1 # 'i' occurs six times
"Indivisibilities" -> 2 # 'i' occurs seven times and 's' occurs twice
"aA11" -> 2 # 'a' and '1'
"ABBA" -> 2 # 'A' and 'B' each occur twice

Ex6.3 - organize strings

Take 2 strings s1 and s2 including only letters from ato z. Return a new sorted string, the longest possible, containing distinct letters,

each taken only once - coming from s1 or s2.

Examples:
a = "xyaabbbccccdefww"
b = "xxxxyyyyabklmopq"
longest(a, b) -> "abcdefklmopqwxy"

a = "abcdefghijklmnopqrstuvwxyz"
longest(a, a) -> "abcdefghijklmnopqrstuvwxyz"

Ex6.4 - isogram

An isogram is a word that has no repeating letters, consecutive or non-consecutive. Implement a function that determines whether a string that contains only letters is an isogram. Assume the empty string is an isogram. Ignore letter case.
isIsogram("Dermatoglyphics") == true
isIsogram("aba") == false
isIsogram("moOse") == false // -- ignore letter case

Implement Functionality

Ex7 - Implement The Following JS Methods -

Implement the following methods -

- Filter
- ForEach
- Map

Using only for(), array and objects (without other js methods)

Ex 8 - Find the Perimeter of a Rectangle

Create a function that takes `length` and `width` and finds the perimeter of a rectangle.

Examples

```
f → 26
indPerimeter(6, 7)
findPerimeter(20, 10) → 60

findPerimeter(2, 9) → 22
```

Notes

- Don't forget to `return` the result.
- If you're stuck, find help in the **Resources** tab.
- If you're really stuck, find solutions in the **Solutions** tab.