

WEEK 3 - ASSIGNMENT 5

BASIC LIST ITERATION

NOTE:

- No need to submit anywhere, just keep track of all the PDF you made in a specific folder.
- Compare your solution with the solution I'll provide, in case of doubts, kindly reach out to me.
- You may get assignment solution in format of PDF or VIDEO solution, depending on the difficulty level.

Q1. When resistors are connected together in series, the same current passes through each resistor in the chain and the total resistance, R_T , of the circuit must be equal to the sum of all the individual resistors added together. That is:

$$R_T = R_1 + R_2 + R_3 \dots$$

Create a function that takes an array of values resistance that are connected in series, and calculates the total resistance of the circuit in **ohms**. The **ohm** is the standard unit of electrical resistance in the International System of Units (SI).

Examples

- `series_resistance([1, 5, 6, 3])` → "15 ohms"
- `series_resistance([16, 3.5, 6])` → "25.5 ohms"
- `series_resistance([0.5, 0.5])` → "1.0 ohm"

Q2. Given a number, return a list containing the two halves of the number. If the number is odd, make the **rightmost number higher**.

Examples

- `number_split(4)` → [2, 2]
- `number_split(10)` → [5, 5]

- `number_split(11) → [5, 6]`
- `number_split(-9) → [-5, -4]`

All numbers will be integers.

You can expect negative numbers too.

Q3. Lists can be mixed with various types. Your task for this challenge is to sum all the number elements in the given list. Create a function that takes a list and returns the sum of all numbers in the list.

Examples

- `numbers_sum([1, 2, "13", "4", "645"]) → 3`
- `numbers_sum([True, False, "123", "75"]) → 0`
- `numbers_sum([1, 2, 3, 4, 5, True]) → 15`

Q4. Create a function that takes a list of numbers and returns the number that's unique.

Examples

- `unique([3, 3, 3, 7, 3, 3]) → 7`
- `unique([0, 0, 0.77, 0, 0]) → 0.77`
- `unique([0, 1, 1, 1, 1, 1, 1, 1]) → 0`

Q5. Create a function that takes a list of positive and negative numbers. Return a list where the first element is the **count** of positive numbers and the second element is the **sum** of negative numbers.

Examples

- `sum_neg([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -11, -12, -13, -14, -15]) → [10, -65]`
 # There are a total of 10 positive numbers.
 # The sum of all negative numbers equals -65.
- `sum_neg([92, 6, 73, -77, 81, -90, 99, 8, -85, 34]) → [7, -252]`
- `sum_neg([91, -4, 80, -73, -28]) → [2, -105]`
- `sum_neg([]) → []`

If given an empty list, return an empty list: []

0 is not positive.