1. Sum Numbers

In a given text you need to sum the numbers. Only separated numbers should be counted. If a number is part of a word it shouldn't be counted.

The text consists from numbers, spaces and english letters

\*\*Input\*\*: A string.

\*\*Output\*\*: An int.

2. Even the last

You are given an array of integers. You should find the sum of the integers with even indexes (0th, 2nd, 4th...). Then multiply this summed number and the final element of the array together. Don't forget that the first element has an index of 0.

3. Right to left

One of the robots is charged with a simple task: to join a sequence of strings into one sentence to produce instructions on how to get around the ship. But this robot is left-handed and has a tendency to joke around and confuse its right-handed friends.

You are given a sequence of strings. You should join these strings into a chunk of text where the initial strings are separated by commas. As a joke on the right handed robots, you should replace all cases of the words "right" with the word "left", even if it's a part of another word. All strings are given in lowercase.

Input: A sequence of strings.

Output: The text as a comma-separated string.

4. Three Words

Let's teach the Robots to distinguish words and numbers.

You are given a string with words and numbers separated by whitespaces (one space). The words contains only letters. You should check if the string contains three words in \*\*succession\*\*. For example, the string "start 5 \*\*one two three\*\* 7 end" contains three words in succession.

Input: A string with words.

Output: The answer as a boolean.

5. First Word

You are given a string where you have to find its first word. When solving a task pay attention to the following points:

* There can be dots and commas in a string.
* A string can start with a letter or, for example, a dot or space.
* A word can contain an apostrophe and it's a part of a word.
* The whole text can be represented with one word and that's it.

Input: A string.

Output: A string.

#### 6. Days Between

How old are you in a number of days? It's easy to calculate - just subtract your birthday from today. We could make this a real challenge though and count the difference between any dates.

You are given two dates as an array with three numbers - a year, month and day. For example: 19 April 1982 will be (1982, 4, 19). You should find the difference in days between the given dates. For example between today and tomorrow = 1 day. The difference will always be either a positive number or zero, so don't forget about the absolute value.

Input: Two dates as tuples of integers.

Output: The difference between the dates in days as an integer.

How it is used: Python has batteries included, so in this mission you’ll need to learn how to use completed modules so that you don't have to invent the bicycle all over again.

Precondition: Dates between 1 january 1 and 31 december 9999. Dates are correct.

#### 7. Count Digits

You need to count the number of digits in a given string.

**Input:** A Str.

**Output:** An Int.

#### 8. Backward Each Word

In a given string you should reverse every word, but the words should stay in their places.

**Input:** A string.

**Output:** A string.

#### 9. Bigger Price

You have a table with all available goods in the store. The data is represented as a list of dicts

Your mission here is to find the TOP most expensive goods. The amount we are looking for will be given as a first argument and the whole data as the second one

**Input:** int and list of dicts. Each dicts has two keys "name" and "price"

**Output:** the same as the second Input argument.

#### 10. Between Markers

You are given a string and two markers (the initial and final). You have to find a substring enclosed between these two markers. But there are a few important conditions:

The initial and final markers are always different.  
If there is no initial marker, then the first character should be considered the beginning of a string.  
If there is no final marker, then the last character should be considered the ending of a string.  
If the initial and final markers are missing then simply return the whole string.  
If the final marker comes before the initial marker, then return an empty string.

**Input:** Three arguments. All of them are strings. The second and third arguments are the initial and final markers.

**Output:** A string.

**How it is used:** for parsing texts

**Precondition:** can't be more than one final marker and can't be more than one initial. Marker can't be an empty string

#### 11. Non-unique Elements

You are given a non-empty list of integers (X). For this task, you should return a list consisting of only the non-unique elements in this list. To do so you will need to remove all unique elements (elements which are contained in a given list only once). When solving this task, do not change the order of the list. Example: [1, 2, 3, 1, 3] 1 and 3 non-unique elements and result will be [1, 3, 1, 3].

**Input:** A list of integers.

**Output:** An iterable of integers.

**Example:**

checkio([1, 2, 3, 1, 3]) == [1, 3, 1, 3]  
checkio([1, 2, 3, 4, 5]) == []  
checkio([5, 5, 5, 5, 5]) == [5, 5, 5, 5, 5]  
checkio([10, 9, 10, 10, 9, 8]) == [10, 9, 10, 10, 9]  
checkio([2]) == []

**How it is used:** This mission will help you to understand how to manipulate arrays, something that is the basis for solving more complex tasks. The concept can be easily generalized for real world tasks. For example: if you need to clarify statistics by removing low frequency elements (noise).

**Precondition:** 0 < len(data) < 1000

#### 13. Second Index

You are given two strings and you have to find an index of the second occurrence of the second string in the first one.

Let's go through the first example where you need to find the second occurrence of "s" in a word "sims". It’s easy to find its first occurrence with a function index or find which will point out that "s" is the first symbol in a word "sims" and therefore the index of the first occurrence is 0. But we have to find the second "s" which is 4th in a row and that means that the index of the second occurrence (and the answer to a question) is 3.

**Input:** Two strings.

**Output:** Int or None

**Example:**

second\_index("sims", "s") == 3  
second\_index("find the river", "e") == 12  
second\_index("hi", " ") is None

#### 14. Frequency Sort

Sort the given iterable so that its elements end up in the decreasing frequency order, that is, the number of times they appear in elements. If two elements have the same frequency, they should end up in the same order as the first appearance in the iterable.

**Input:** Iterable

**Output:** Iterable

**Precondition:** elements can be ints or strings