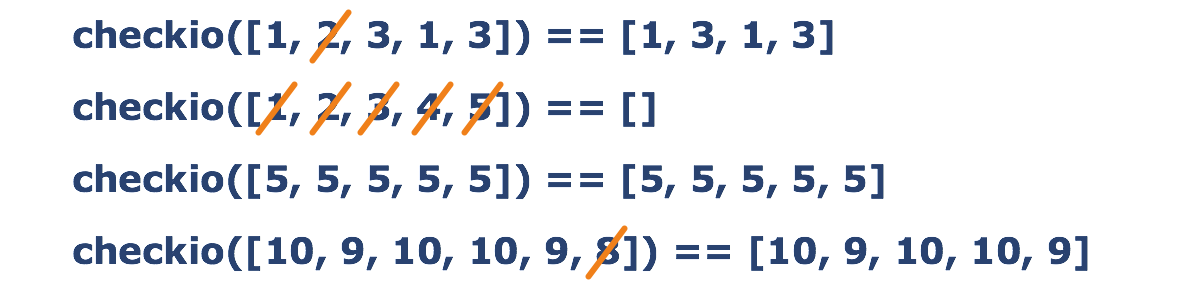
**Workshop**

**Question 1.** 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.

**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).

**Türkçesi** : Size boş olmayan integerlardan oluşan bir liste veriliyor. Bu görev için içinde unique eleman olmayan bir liste return etmeniz isteniyor. Bunu yapmak için size verilen listede sadece bir kere geçen elemanları kaldırmanız gerekmektedir. Bunu yaparken listenin sırasını değiştimemeniz gerekmektedir. Lütfen örnekleri inceleyiniz.

**Question 2.** 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.

Example:

1. checkio("Hello World hello") == True
2. checkio("He is 123 man") == False
3. checkio("1 2 3 4") == False
4. checkio("bla bla bla bla") == True
5. checkio("Hi") == False

**How it is used:**This teaches you how to work with strings and introduces some useful functions.

**Türkçesi** : Size boşluklarla ayrılmış kelimelerden ve sayılardan oluşan bir string veriliyor. Kelime yanlızca harlerden oluşur. Sizden ard arda üç kelime varmı yokmu onu control etmeniz istenmektedir. Lütfen örnekleri inceleyiniz.

**Question 3.** In a given text you need to sum the numbers while excluding any digits that form part of a word.

The text consists of numbers, spaces and letters from the English alphabet.

**Input:**A string.

**Output:**An int.

**Example:**

sum\_numbers('hi') == 0

sum\_numbers('who is 1st here') == 0

sum\_numbers('my numbers is 2') == 2

sum\_numbers('This picture is an oil on canvas '

'painting by Danish artist Anna '

'Petersen between 1845 and 1910 year') == 3755

sum\_numbers('5 plus 6 is') == 11

sum\_numbers('') == 0

**Türkçesi** : Sizden kelimenin bir parçası olmayan sayıları toplamanız istenmektedir. Lütfen örnekleri inceleyiniz.

**Question 4.** 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.

For an empty array, the result will always be 0 (zero).

**Input:**A list of integers.

**Output:**The number as an integer.

**Example:**

checkio([0, 1, 2, 3, 4, 5]) == 30

checkio([1, 3, 5]) == 30

checkio([6]) == 36

checkio ([]) == 0

**Türkçesi** : Size integer lardan oluşan bir liste veriliyor. Sizden çift indexe sahip olan elemanları toplamanızı daha sonra bu toplamı listenin son elemanı ile çarpmanız isteniyor. Lütfen örnekleri inceleyiniz.

**Question 5.** 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.

**Example:**

left\_join(("left", "right", "left", "stop")) == "left,left,left,stop"

left\_join(("bright aright", "ok")) == "bleft aleft,ok"

left\_join(("brightness wright",)) == "bleftness wleft"

left\_join(("enough", "jokes")) == "enough,jokes"

**How it is used:**This is a simple example of operations using strings and sequences.

**Türkçesi** : Size virgül ile ayrılmış bir dizi string veriliyor. Siz bunları birleştirip bir text yapacaksınız. Bunu yaparken kelimenin bir parçası dahi olsa bütün “right” yazan yerleri “left” ile değiştirmeniz isteniyor. Lütfen örnekleri inceleyiniz.

**Question 6.** 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, one/multiple dot(s) or space(s).
* 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.

**Türkçesi** : Sizden verilen bir stringin ilk kelimesini bulmanız isteniyor. Bunu yaparken aşağıdaki hususlara dikkat etmeniz isteniyor.

Stringin içinde noktalar ve virgüller olabilir.

String bir harfle veya bir veya birden fazla noktayla veya boşlukla başlayabilir.

Kelime bir apostrof içerebilir ve bu kelimenin bir parçası olabilir.

Bütün bir text bir kelime gibi gösterilmiş olabilir.  
Lütfen örnekleri inceleyiniz.

**Examples:**

first\_word("Hello world") == "Hello"

first\_word(" a word ") == "a"

first\_word("don't touch it") == "don't"

first\_word("greetings, friends") == "greetings"

first\_word("... and so on ...") == "and"

first\_word("hi") == "hi"

first\_word("Hello.World") == "Hello"

**Question 7.** 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.

**Türkçesi** : Yaşınız gün hesabı olarak acaba kaç? Bunu hesaplamak kolay. Sadece bu günün tarininden doğduğunuz günün tarhini çıkaracksınız. Bu zorlayıcı olabilir. Size üç sayıdan oluşan bir dizi halinde iki tarih veriliyor. Ör 19 Nisan 1982 (1982,4,19). Sizden gün olarak verilen tarihler arasındaki farkı bulmanız isteniyor. Lütfen örnekleri inceleyiniz.

**Examples:**

days\_diff((1982, 4, 19), (1982, 4, 22)) == 3

days\_diff((2014, 1, 1), (2014, 8, 27)) == 238

days\_diff((2014, 8, 27), (2014, 1, 1)) == 238

**Question 8.** You need to count the number of digits in a given string.

**Input:**A Str.

**Output:**An Int.

**Türkçesi** : Sizden verilen bir stringdeki rakamların adedini bulmanız isteniyor. Lütfen örnekleri inceleyiniz.

**Example:**

count\_digits('hi') == 0

count\_digits('who is 1st here') == 1

count\_digits('my numbers is 2') == 1

count\_digits('This picture is an oil on canvas '

'painting by Danish artist Anna '

'Petersen between 1845 and 1910 year') == 8

count\_digits('5 plus 6 is') == 2

count\_digits('') == 0

**Question 9.** In a given string you should reverse every word, but the words should stay in their places.

**Input:**A string.

**Output:**A string.

**Türkçesi** : Sizden verilen bir stringdeki kelimelerin tersten yazmanız isteniyor. Yalnız kelimelerin yeri değişmeyecek. Lütfen örnekleri inceleyiniz.

**Example:**

backward\_string\_by\_word('') == ''

backward\_string\_by\_word('world') == 'dlrow'

backward\_string\_by\_word('hello world') == 'olleh dlrow'

backward\_string\_by\_word('hello world') == 'olleh dlrow'

**Question 10.** 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.

**Türkçesi** : Size Bir mağazadaki bütün ürünlerin bilgilerinin olduğu bir tablo veriliyor. Veri listelerden oluşan bir dictionary. Buradaki göreviniz en pahalı ürünleri bulmanız. Fonksiyondaki ilk argüman kaç ürün istediğimizi , ikinci argüman ise bütün bilgilerden oluşuyor. Lütfen örnekleri inceleyiniz.

**Example:** bigger\_price(2, [{"name": "bread", "price": 100}, {"name": "wine", "price": 138},

{"name": "meat", "price": 15}, {"name": "water", "price": 1}]) ==

[ {"name": "wine", "price": 138}, {"name": "bread", "price": 100}]

bigger\_price(1, [ {"name": "pen", "price": 5}, {"name": "whiteboard", "price": 170}

]) == [{"name": "whiteboard", "price": 170}]