

If you have 50 different plug types, appliances wouldn't be available and would be very expensive. But once an electric outlet becomes standardized, many companies can design appliances, and competition ensues, creating variety and better prices for consumers.

-- Bill Gates

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].

**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]**

**Input:** A list of integers.

**Output:** An iterable of integers.

**Example:**

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

You can find out more about Python arrays in [one of our articles featuring lists, tuples, array.array and numpy.array](#).

**Precondition:**

$0 < \text{len}(\text{data}) < 1000$