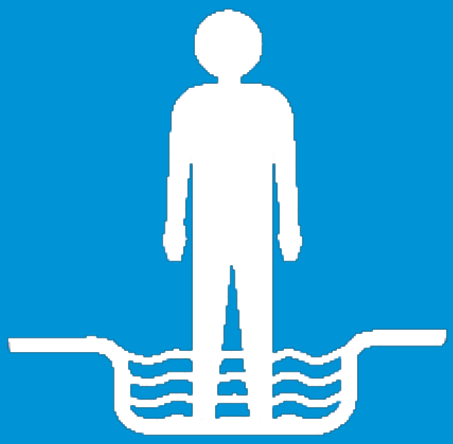
{EPITECH}

PRE-POOL

DAY 05



PRE-POOL

OH MY GIT!
In addition to the tasks below, you must go as far as possible in this game.
Work on it as soon as you have a bit of time, or whenever you need a break in you day!

Lists creation and browsing

Task 1.1

Create a list that contains 5 numbers and print the first one.

Task 1.2

Display the last element of your list.

Your code must be functional whichever number of elements the list contains.

Task 1.3

Add a 6th element in your list.

Task 1.4

Display all the elements of your list.

Task 1.5

Delete the last element and display all the remaining elements.

> Your code must be functional whichever number of elements the list contains.

Task 1.6

Add an element at the beginning of the list and display all its elements.

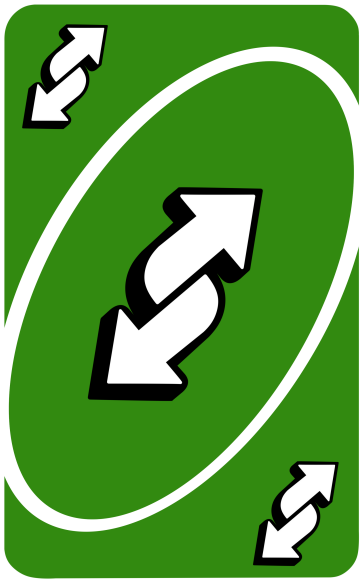
Task 1.7

Display the sub-list from the 2nd to the 5th elements.

> Can you do it in one line?

Task 1.8

Reverse the list by creating a new list with the same elements, but starting from the end.
Display all the elements of this new list.



Task 1.9

Display one element out of two of the list.

Task 1.10

Add 17 elements at the end of your list.

Please, do not do it in 17 lines...

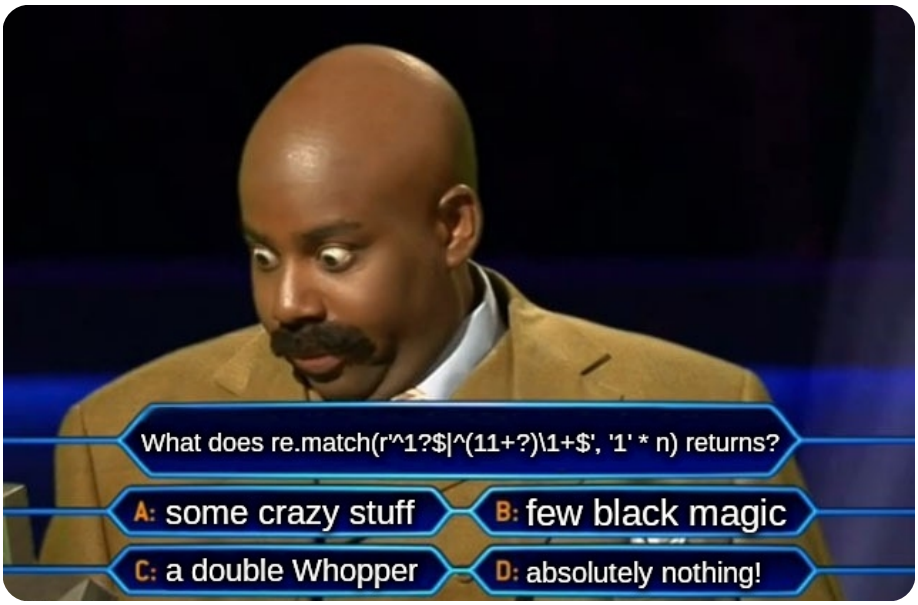
Task 1.11

What does the following code make?

```
my_first_list = [4, 5, 6]
my_second_list = [1, 2, 3]
my_first_list.extend(my_second_list)
```

Same with:

```
my_first_list = [7, 8, 9]
my_second_list = [4, 5, 6]
my_first_list = [*my_first_list, *my_second_list]
```



Operations on lists

Task 2.1

Create a list of 10 numbers.
Print the result of the multiplication of all elements of this list.

Task 2.2

Test this code and try to explain it: `[x + 10 for x in [3, 2, 6, 7, 1, 4]]`

Task 2.3

Test this code and try to explain it: `list(map(lambda x: x * x, [3, 2, 6, 7, 1, 4])`

Task 2.4

Browse the list and display both the smallest and the biggest elements.

Task 2.5

Display all the elements smaller than 7.

Task 2.6

Sort your list in descending order.

Task 2.7

Test this code and try to explain it `[x // 2 if x % 2 == 0 else x * 2 for x in [42, 3, 4, 18, 3, 10]]`

Task 2.8

Test this code and try to explain it: `list(filter(lambda x: x > 10, [42, 3, 4, 18, 3, 10]))`

Task 2.9

Test this code and try to explain it: `[*enumerate([42, 3, 4, 18, 3, 10])]`

Task 2.10

Test this code and try to explain it:

```
first_name = ["Jackie", "Bruce", "Arnold", "Sylvester"]
last_name = ["Stallone", "Schwarzenegger", "Willis", "Chan"]

magic = [*zip(first_name, last_name[::-1])]

print(magic[0])
print(magic[3])
print(magic[1][0])
print(magic[0][1])
print(magic[2])
```

CHALLENGE

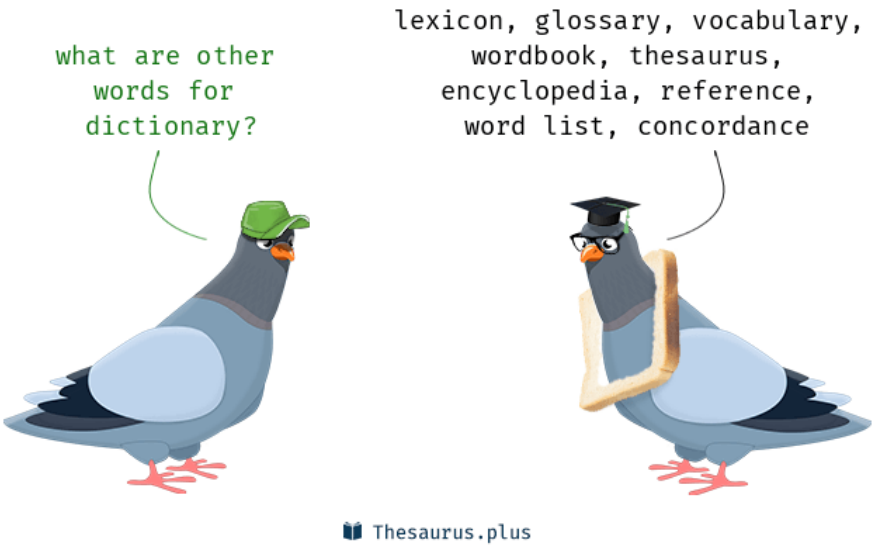
Create a list of 1 000 000 random integers and sort it as fast as possible.

If you want to know precisely how long the execution of your program lasted, you can put the code `startingTime = time.time()` at the beginning and `duration = time.time()- startingTime` at the end.

Dictionaries

Task 3.1

Create a dictionary stored in a `student` variable.
This dictionary must contain 2 key/value pairs.
The keys must be “name” and “academic_year”.
The associated values are up to you, but please keep them coherent!



Task 3.2

In your student dictionary, add a `units` key.
The value associated to `units` is an array of 3 elements.
Each element of this array is a dictionary of 3 keys: `name`, `credits`, `grade`:

The value associated with `name` can be one of:

- ✓ Web Development
- ✓ Network and System Administration
- ✓ Java

The value associated with `credits` is a strictly positive integer.
The value associated with `grade` is a letter ranging from A to E.

Task 3.3

Create a new dictionary named `grade_weight_mapping`.
This dictionary contains 5 keys (“A”, “B”, “C”, “D”, “E”) and their respective values (4, 3, 2, 1, 0).

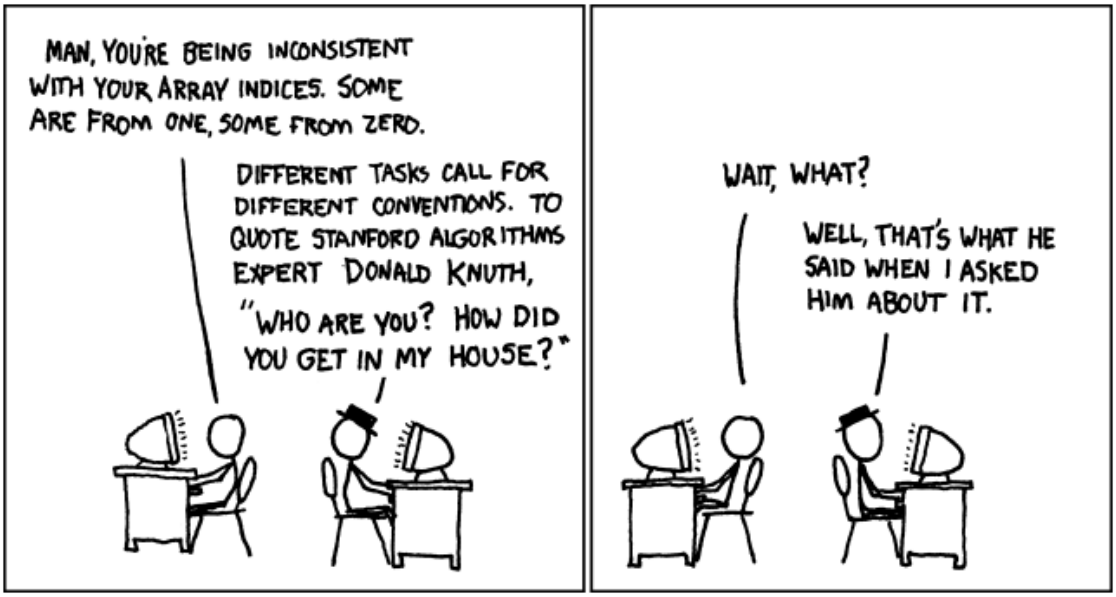
In your `student` dictionary, add a `total_credits` name. Give it the correct value according to the amount of credits you’ve given to each units in the previous task.
Also add a `GPA` key and give it the correct value.

GPA or Grade Point Average is a number between 0 and 4. It is the average grade obtained, weighted by the number of credits.

Task 3.4

Create an array and store at least 3 `students` (as defined in previous tasks) in it.
Write some code to:

- ✓ sort this array by student’s name in alphabetical order
- ✓ sort this array by GPA in both ascending and descending order



Lets' go further

Task 4.1

Let's consider a list of names (the ambassador's banquet guests).
Write a program that displays:

- ✓ "welcome in" if a given name belongs to this list ;
- ✓ and "get lost!" otherwise.

Task 4.2

Write a program that deletes all the duplicated elements in a list.

Task 4.3

Let consider a list of meetings. Each meeting is a list containing the day, the time of the meeting and the name of all the participants.
For instance `["Monday, "3:30 PM", "Joe", "Samantha"]`.
Write a program that, given a name, displays all the meetings in which this person is involved.

Task 4.4

Humm… you seem like a fast scorer!
Wander in the halls and find some desperate colleagues that need code reviewing.
And of course, review their code.

You may even try a bit of peer-coding.

v 1.0.2