OOP magic functions – exercises

# Exercise 1

This exercise deals with the Person class we’ve defined earlier.

## Part I

Define magic functions for the following functionalities:

* String representation that includes the main attributes
* Equality that is based on the ID number
* Comparison that is based on the age (ignore *is\_alive*)

## Part II

Create an instance of the class and test all the methods. Try to simulate the life story of someone you know. Print relevant attributes after every method, so that you’ll monitor the results.

# Exercise 2

This exercise deals with the Queue class we’ve defined earlier.

## Part I

Define magic functions for the following functionalities:

* *\_\_len\_\_()* instead of the method *get\_len()*
* Item getter (*\_\_getitem\_\_()*) that for the key *k* returns the element in the “real” k-th place. If there is no k-th element, then the method returns *None*. For example, *obj[1]* returns the element that is “first in line”.
* String representation that includes the content of the first three elements (or a part of them if there are less than three elements)
* Comparisons that are based on the length of the queues
* Addition of queues that concatenates the queues (Since the order of self and other is of significance, you should use \_\_radd\_\_() instead of \_\_add\_\_(). The relevant documentation is [here](https://docs.python.org/2/reference/datamodel.html#object.__radd__)).

## Part II

The file “queue2.txt” contains a list of patients by the time of their arrival to a clinic, but now there are two doctors, and the patients always join the shorter queue. Once in a while one of the doctors “calls” the patient who is first in the associated line. Apply the class *Queue* to answer who is the patient that suffered the longest line.