4. Tutorial/practical exercises

Exercise 1

Using lists, write an efficient function that finds all primes numbers less than 10000.

Exercise 2

Write a function that takes a list of words, then produces a list whose elements are words that begins with the same letter. The lists should be sorted.

For example, if the input is `['apple','bed','banana','ape','cow']`, the output should be `[['ape','apple'], ['banana','bed'],['cow']]`.

Exercise 3

Write a function that receives a list strings that can be representations of integers, floats, dates, or unformatted strings, then produces fours lists: one for integers, one for floats, one for dates, and one for unformatted strings. The lists should contain the right types.

For example, if the input is `["1","True","0","2.2",05/04/2025","name"]` then the output should be `[[True], [1,0],[2.2],['05/04/2025'],["name"]]` (you should use the `datetime` module for dates).

Exercise 4

Write queues primitives using two stacks.

Exercise 5

A multi-queueing system is composed of n queues, such that q_i has priority i. Element are queued with priorities 1, 2, …, n. Dequeueing is done from the queue the with highest priority if it is not empty. Write a module `multi_queuing.py` that implements this system by defining:

1. A function that creates a multi-queue (with n queues)
2. A function that enqueues an element by specifying its priority
3. A function that dequeues an element.

Exercise 6

A list consists of numbers with possibly repeating values; we consider the order irrelevant.

1. Write a function that converts the list into a dictionary, with keys representing the list's items and values representing the number of occurrences. For instance, the result for `[2,3,2,1,1,1,3]` should be `{1:3,2:2,3:2}`.
2. Write the inverse function that converts a dictionary to a list.
3. Write a function that checks whether two lists are equal. + We define a subtraction operation for two lists. For example, `[2,3,2,1]` - `[1,2,3]` equals `[2]` (removing duplicated components based on their count in the second list). Write a function that performs this operation.

Exercise 7

Students are characterized by an ID, a name, an age, and an address. A course is characterized by its ID and a title. A student is enrolled in a course.

1. Write dataclasses that correspond to this situation.
2. Write a function that enrolls a student in a course.
3. Write a function that finds the students enrolled in a course.

Exercise 8

Consider a point characterized by its x and y coordinates (don't use the `numpy` module).

1. Write a dataclass that represents a point.
2. Write a function that computes the distance between two points.
3. Consider a point $P = (x, y)$ and a radius r. A point $Q = (x', y')$ is within the circle of center (x, y) if the distance between P and P' is less than or equal to r. Write a function that determines if a point Q is within the circle of center P and a radius Q.
4. Write a function that takes a set of n points and a radius r, then sorts the points based on the number of points contained within a circle whose center is each point and radius r.
5. Write a function that uses the last function, then draws circles until all points are contained in at least on circle.

Hints for drawing:

- Use the `matplotlib` module (`pyplot` and `patches`).
- Create a subplot with `py fig, ax = plt.subplots()`
- To draw a circle, use `py ax.add_patch(patches.Circle((x,y),radius,color='b', fill=False))`
- To show the figure, use `py plt.show()`
- Possibly, you can use `ax.set_xlim(...)` and `ax.set_ylim(...)` to define the limits of the figure.