Computer Systems Modeling and Verification (USEEN1)

List comprehension

Exercises taken from the *Python Programming Primer*[1] (and updated with type hints).

Textbook *Introduction to Python for Computational Science and Engineering* (2022), Chapter 8.

Corresponding lecture slides from *Computational Science and Engineering in Python*:

- Dictionaries
- List comprehension

Exercises. Define the following functions.

1. A function `positive_places(f: Callable[[float], float], xs: Sequence[float]) -> list[float]` that takes as arguments some function `f` and a list of numbers `xs` and returns a list of those-and-only-those elements `x` of `xs` for which `f(x)` is strictly greater than zero. This task was given already as a training exercise in lab 6 where we also provide some input and output examples.

   In this lab, we ask that you write the same function *without* usage of a `for` or `while` loop to practice list comprehension or the use of `filter`.

2. Write a function `reverse_dict[K, V](d: dict[K, V]) -> dict[V, K]` that takes a dictionary `d` as the input argument and returns a dictionary `r`. If the dictionary `d` has a key `k` and an associated value `v`, then the dictionary `r` should have a key `v` and a value `k`. Not that this function is only expected to work for dictionaries where each value occurs at most once (although you do not need to check for this).

[1] *Python Programming Primer*, Hans Fangohr *et al.* University of Southampton (2016)