University of Oldenburg
Python Programming in Energy Science, WiSe 2023-2024
Balthazar Sengers, balthazar.sengers@iwes.fraunhofer.de

Assignment 2

Library imports are generally not allowed for this sheet, please solve all tasks using pure vanilla Python.
Exception: `numpy.random.randint`

E1. Data analysis

In this exercise we read and analyse the provided data file "winddata.csv". Assume that the first row of the file lists the column names. The first column is a time string and the second and third columns represent the x and y components of the wind vector at 100 m height in m/s, respectively. All data are seperated by commas.

- Write a function *read_csv(filepath)* that reads a file at the given file location into a list of lists.
- Calculate the mean wind vector and its magnitude based on the above mentioned data file.
- Calculate the mean wind speed and its standard deviation. Write your own functions, don't use libray imports!
- Using the built-in `any()` and `all()` functions, check whether all wind speeds are positive (hence no filler values like -999 exist), and whether wind speeds larger than 25 m/s occur.
- Find the minimum and maximum wind speed in this data set.
- In bins of 1 m/s, count how many values fall in each bin between the minimum and maximum wind speed.
- What is the longest period of time that the wind speed was continuously above a threshold of 10 m/s? Repeat for thresholds of 15 and 25 m/s.

E2. Database

Let's consider the following table of persons, ages and favourite colors:

Name	Age	Color
Julia	3	Green
Jim	32	Red
Marco	16	Green
Denise	23	Blue
Paula	28	Red
Louis	42	Yellow

- Create a Python *dict*, with entries of type *dict*, that contains the above data, and print it.
- Let the user pick a name, and print the selected person's data. Throw a *KeyError* if the name cannot be found.
- Get a color from the user and print all names of persons that have that favourite color. Throw a *ValueError* if no name is associated with that color.
- Get a minimal and a maximal age, and find all persons within that range. This time print some message if no person could be found (no error).

E3. Poetry generator

In this exercise we create random lines of poetry, all with structure

The [subject] [verb] [object], [comment]

- Create a dict, with entries *subject, verb, object, comment*, each associated with a list of strings. Fill the lists with data that makes sense for the above sentence structure (at least 5 entries for each category). The *comment* should finish with a dot, comma, exclamation mark or question mark.
- Write a *generator* (see lecture 3!) that yields a new random sentence based on the above created dict every time it is called. Use numpy's `randint` function for the random choices (cf. Assignment 1 and its helper notebook).

- Create a poem with 3 such random sentences (each in its own line), and print it.

E4. Classic phone book

This exercise presents one of the most classic examples of the use of a dictionary: the phone book. Starting with an empty phone book, we give the user the add new contacts, display all existing contacts or search for specific ones.

- Write a function that adds a new contact to the phone book. Each contact should have a first name and a corresponding phone number. The first new is unique and should not be overwritten.
- Write a second function that lists all existing contacts. Only the names should be listed, not the phone numbers.
- Write a third function that searches for a contact when the name is given. It should then display the phone number of that person.
- Now initialize an empty phone book and fill it with at least 5 names and phone numbers, using the function developed above. After entering each new phone number, the program should ask what to do next (hence which of the functions to call) or whether to quit.

E5. String analysis

- Read the provided file *holy_grail.txt* into a string variable.
- How many lines and, how many words and how many characters are in this text?
- How often do the word `python`, the word `Lancelot` and the combination `holy grail` appear in the text, irrespective of lower/upper case?
- Replace all exclamation and question marks by dots. Then replace the word `Python` by `Cobra` in the whole text. Among all sentences, defined by ending with a dot, find those that contain the word `Cobra` and print them.

E6. Fibonacci sequence

- Write a function that prints the Fibonacci sequence of 10 Fibonacci terms.
- Adapt your function such that instead of printing the sequence directly, it returns a list containing the sequence.
- Assume that the Fibonacci sequence you just generated contain the radii of a set of circles. Using the build-in `map()` function, calculate and print the area of these circles. Use $\pi = 3.1415$.
- Instead of using a function, use a generator construction to print the Fibonacci terms. Look at the lecture slides for inspiration.