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

Assignment 1

For this sheet no library import is allowed. Please solve all tasks using pure vanilla Python.
Exceptions are given in the assignments

E1. Number guessing

Write a small programm (in a single notebook cell), where the user inputs a number between 0 (included) and 100 (included), and Python tries to guess the number randomly.

- Let the program do a first random guess within the full interval between 0 and 100. For this, load the module with `from numpy.random import randint`
- Then compare the guessed and the goal number, and print the guessed number together with remarks 'too low' or 'too high', or 'match!'.
- Adopt the range of possible numbers. Let the programm guess again, proceed as explained above. Continue until the number has been found.
- Print the total number of needed guesses.

E2. Vector magnitudes

A vector in three dimensions has three components *x, y, z*. It is your decision how to realize such a data structure in the following: Use a *tuple*, a *list*, a *dict* or write a *class* (check the lecture and the helper notebook for all of those!).

- Create a list of 10 such vectors, filled with random integer (again use `from numpy.random import randint`) coordinates *x, y, z* (between 0 and 30, both included).
- Write a function that calculates the magnitude $\sqrt{x^2+y^2+z^2}$ for a single vector. Apply the function to each of the vectors, and print the vector along with the magnitude.
- Find the vectors with the smallest and the largest magnitude, and print them.

E3. Wind turbine wake

- Assume wind with uniform wind speed $u_0 = 9$ m/s is hitting a 6 MW wind turbine with rotor diameter $D = 154$ m, located at coordinate $x = 0$ m. For this wind speed the thrust coefficient below is $c_t = 0.763$.
- According to the paper [1], the wind speed u at a distance x behind the turbine can be modelled as

$$u(x) = u_0\sqrt{1-\frac{c_t}{8}\left(\frac{D}{\sigma(x)}\right)^2}\,, \qquad \sigma(x) = kx + \frac{1}{4}\sqrt{\beta}D\,, \qquad \beta = \frac{1+\sqrt{1-c_t}}{2\sqrt{1-c_t}}\,, \qquad k = 0.02\,.$$

After how many meters behind the rotor has the wind speed recovered to at least 8.95 m/s? Define functions for $\sigma(x)$ and $u(x)$ in Python and find out by increasing x in a loop!

E4. Geometric series

- Write a function that explicitely calculates the sum $\sum_{k=0}^{N-1} x^k$ for any given integer N and number x.
- Write another function that calculates the closed form of the geometric series $\frac{1-x^N}{1-x}$.
- Check if for $0 < x < 1$ the two functions give the same result by testing all N up to 50. Test this for four different values x of your choice.
- Explain possible deviations, if you observe any!

E5. Graph intersect

Consider two functions: $f(x) = sin(x)$ and $g(x) = cos(x)$.

- Define these two function as python functions, using an angle (in degrees) as input. Use `module math`.
- Find out at what angles the two graphs intersect. For this purpose, loop over all degrees between 0 and 360 in 1 degree increments.

Now consider the following two functions: $f(x) = x^2 + 2x - 5$ and $g(x) = 2x - 1$ for the range [-5, 5].

- Again define these functions as python functions and find out where they intersect.
- These two functions intersect twice. Find the integrated area between the functions between these two points. Use the trapezoidal rule with 100 steps.

E6. Linear regression

In this exercise we will perform a linear regression on a cloud of point using Ordinary Least Squares. The goal is to find the coefficients a and b in the function $y = ax + b$.

- Copy the following code to generate this cloud of points:

```
def generate_points():
    import numpy as np
    x = np.linspace(0, 10, 1000)
    noise = np.random.normal(0, 3, len(x))
    a,b = np.random.randint(1, 5), np.random.randint(-3, 3)
    y = a * x + b + noise
    print(f'Original␣function:␣y␣=␣{a}x␣+␣{b}')
    return x,y
```

- Define a function to find the coefficients a and b. Do this using Ordinary Least Squares:

$$a = \frac{\sum_i^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_i^n (x_i - \overline{x})^2} \ , \qquad b = \overline{y} - a\overline{x} \ .$$

- Explain the differences between the original function from `generate_points()` and the fitted function. How could the fitted coefficients get closer to the original ones?

E7. Christmas tree

- Draw the following Christmas tree using the print-statement:

```
    *
  * * *
* * * * *
    |
```

- If you haven't already done so, write a function that prints this Christmas automatically. The only input to this function is the height of the tree (4 in this example). Also draw one that has height = 10.
- This bigger tree does not look very stable. The tree trunk should be taller and wider. Adapt your function so that the trunk size is automatically adapted to the size of the tree. Both the height and the width of the tree trunk should be approximately one-third of the total three size. For this example, that means that the trunk should be $10/3 \approx 3$ high and wide (3 times |). Test whether you function works by drawing trees of different sizes.

References

[1] M. Bastankhah and F. Porté-Agel. A new analytical model for wind-turbine wakes. Renewable Energy, 70:116–123, 2014.