# TIE2030

# Programming Methodology with Python

# Laboratory Exercise (LAB-III)

**Date submission due: Thursday 4 November 2021**

**Grading: Your program will be graded out of 100 marks and the final weight of this assignment is 20%. The guidelines explained in the template will also carry marks. So please adhere to the guidelines.**

## Question 1: [20 marks]

You are given a file **sequences.txt**, containing 100 DNA sequences (one sequence per line). Read the sequences from the file and do the following:

a. Measure the length of each sequence. Write the sequence and its length to an output file **seq_length_output.txt** with meaningful messages.
b. Plot a histogram that captures the length distribution. Clearly present your chart with all the required information, title, and axis labels.
c. Measure the total time taken to execute your program. Write the result to your output file **seq_length_output.txt** with a meaningful message.

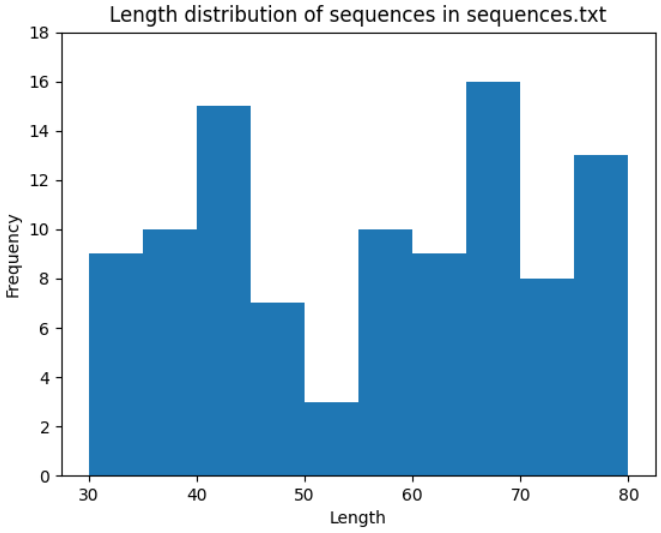
In your report, paste the screenshot of your code, your results, and your histogram.

### Sample output:

```
Sequence 0: AGGACTAACGCATCGCACACGGGGTTCCCTATTGTCCCGCCTGCGAGTTGAGCCGGGGACTTGTT
Length of sequence 0: 65
Sequence 1: CCGCTAGCCGAATAAATAAAACCAGCATCC
Length of sequence 1: 30
...
Sequence 99: GTGGAAAAATGAATCCGACCGTAAGCCACCAAGCTTATA
Length of sequence 99: 39

Processing time: 10.00 seconds
```

**Note:** The distribution shown in the sample histogram is NOT the correct result for the data in **sequences.txt**. It is only shown as a sample output.




## Question 2: [40 marks]

Reuse the data file **sequences.txt** as given in Question 1. Read the strings from the data file and store their lengths in a list. Do the following.

a. Concatenate the strings using an algorithm as described below.

**Sequence Concatenation Algorithm:**

Use this **example** to understand the logic.

Let us say the sequence lengths are captured in a list: [3, 4, 7, 2, 5, 9, 12, 8, 9, 10, 2, 3]. We will identify the sequences that can be concatenated as follows. Take a number from the list from the start of the list and keep adding its adjacent numbers as long as the total count does not exceed a given THRESHOLD.

In our example, THRESHOLD = 15. The output will be as follows:

```
Starting from 3: 3+4+7 = 14 (number of entries: 3, total value: 14,
sequences are: 0, 1, 2)
Then, start with 2: 2+5 = 7 (number of entries: 2, total value: 7,
sequences are: 3, 4)
Then, start with 9: 9 = 9 (number of entries: 1, total value: 9,
sequences are: 5)
Then, start with 12: 12 = 12 (number of entries: 1, total value: 12,
sequences are: 6)
Then, start with 8: 8 = 8 (number of entries: 1, total value: 8,
sequences are: 7)
Then, start with 9: 9 = 9 (number of entries: 1, total value: 9,
sequences are: 8)
Finally, start with 10: 10+2+3 = 15 (number of entries: 3,
total value: 15, sequences are: 9, 10, 11)
```

Now, in your implementation, use the data in **sequences.txt** and **THRESHOLD = 200.** Write your results with meaningful messages to an output file **string_concat_output.txt**

b. Store the number of entries you used to concatenate the strings (in part a) to a list **num_of_entries**. Compute and print the number of occurrences of each value in the above list. Write your results with meaningful messages to your output file **string_concat_output.txt**

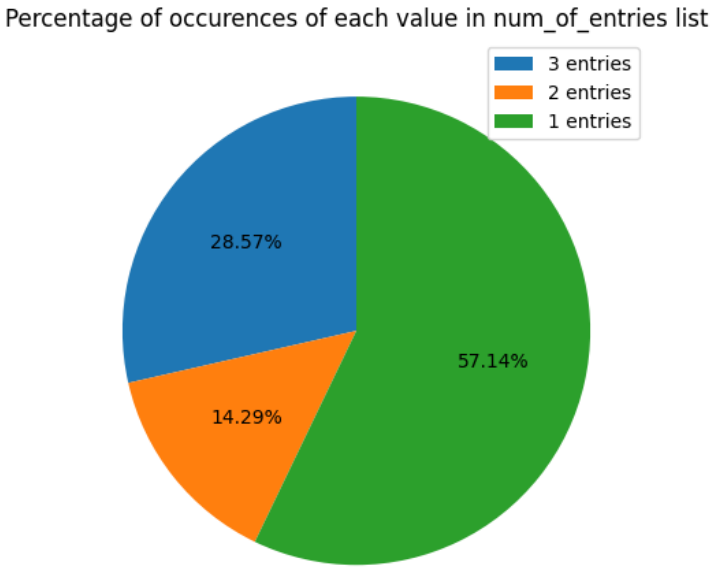
Example: For the example used in part (a), your **num_of_entries** list is [3, 2, 1, 1, 1, 1, 3]. The number of occurrences of each value in the above list are:

```
Number of occurrences of 3 entries: 2
Number of occurrences of 2 entries: 1
Number of occurrences of 1 entries: 4
```

**Note:** Your output file should contain the output of both part (a) and part (b).

c. Plot a pie chart that captures the percentage of occurrences of each value in your **num_of_entries** list. Clearly present your chart with all the required information, title, and legends.

**Sample pie chart** (using data in the example described in part a):




d. Now, change **THRESHOLD = 150** and repeat parts (a)-(c) but write your outputs to **string_concat_output_test2.txt**. You should not change anything in your code but the THRESHOLD value and the output file name, and you should be able to get the correct results for this new test case.

In your report, paste the screenshot of your code, your results, and your pie charts for all the test cases (THRESHOLD = 200 and THRESHOLD = 150).

## Question 3: [40 marks]

You are given an input file **coefficients.txt**. Each line contains three real numbers (a b c) separated by a space character and they are the coefficients of a quadratic equation ($f(x) = a * x^2 + b * x + c$). Identify the equations that have real roots and write your results with meaningful messages to an output file **coefficients_output.txt**.

**Note:** A quadratic equation $f(x) = a * x^2 + b * x + c$ has real roots if:

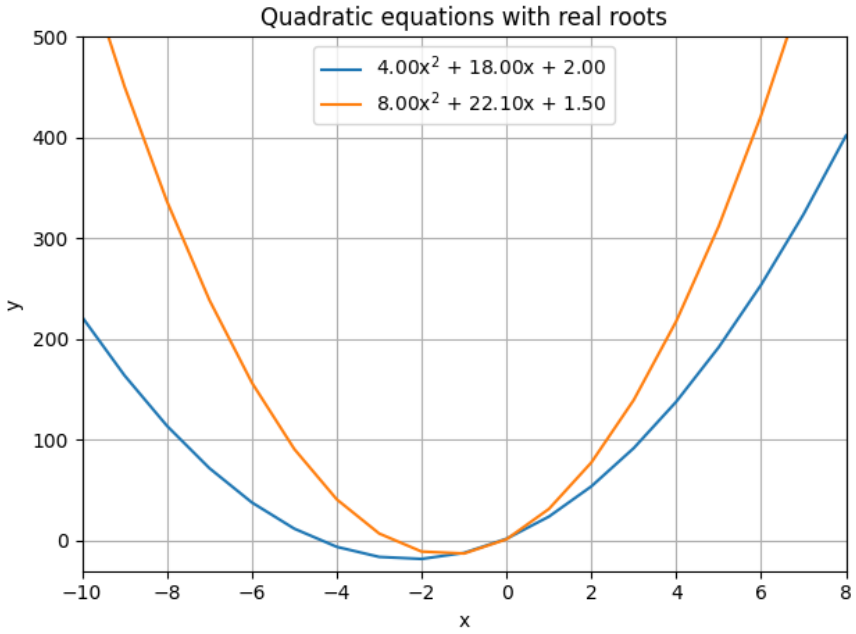
$$b^2 - 4 * a * c \geq 0$$

Next, consider ANY TWO equations that have real roots and **plot them on a single canvas**. Clearly present your plot with all the required information and legends.

### Sample output:

**Note:** This is only a sample output, NOT the correct results for the given input data.

```
Quadratic equation 1 with coefficients (4.00, 18.00, 2.00) has real roots.
Quadratic equation 4 with coefficients (8.00, 22.10, 1.50) has real roots.
Quadratic equation 5 with coefficients (0.25, 19.00, 0.50) has real roots.
```




In your report, paste the screenshot of your code, your results, and your plot.