Worksheet 01: Files and Excepitons

Exercise 1 Write a Python code that copies a text file, but triples all the spaces between the words.

Exercise 2 Write a Python code that allows you to easily create and read a text file.

Required work: - Your program will first ask the user to enter the name of the file. - Offer the user the choice to either save new lines of text or display the contents of the file. - If the user chooses to save new lines of text, allow him to enter successive lines using just the `<Enter>` key to separate them, and save them to the file. - To finish entering text, he just need to enter a blank line (i.e., press the `<Enter>` key alone). - If the user chooses to display the contents of the file, show the lines of the file separated from each other in the most natural way possible (without end-of-line codes).

Exercise 3 (Text Analysis) In this exercise, you will write a program that analyzes a text file called "input.txt". The program should perform the following tasks:

1. Read in the contents of the file and display them on the screen.
2. Count the number of occurrences of each word in the file and display the results on the screen.
3. Find the most common word in the file and display it on the screen.
4. Find the longest word in the file and display it on the screen.
5. Write a new text file called "output.txt" that contains the same data as "input.txt", but with all the vowels removed.

Exercise 4 Suppose there is a typed file called "exam.txt" that contains records related to candidates in a competition. Each record is composed of: ID, NAME, FIRST NAME, AGE, DECISION (a type containing the following identifiers: admitted, rejected, deferred), separated by semicolons (;).

Required work:

- Define the `add()` function that allows filling in the data related to the candidates in the `exam.txt` file.
- Define the `success()` function that allows creating the `success.txt` file containing data related to admitted candidates.
- To prioritize admitted candidates who are under 30 years old, create the `waiting()` function that produces a new file called `waiting.txt` containing data related to admitted candidates

who are over 30 years old. A line in the `waiting.txt` file includes the ID, NAME, and FIRST NAME of a candidate separated by semicolons (;).

- Define the `statistics(dec)` function that allows returning the percentage of candidates for the decision `dec` (admitted, rejected, and deferred). For example:

  $$\text{percentage of admitted} = (\text{Number of admitted} \ / \ \text{Total number of candidates}) \times 100.$$

- Define the `delete()` function that will delete candidates over 30 years old from the `success.txt` file.