LAB 06: STRUCTS and FILESTREAMS

1 In-Class

A. **STRUCT**

1. Use **struct** to represent a timeline that consists of day, month, year, hour, minute, and second. Write functions to complete the following requirements.
   - Construct a timeline from elements entered by user and display the timeline to the screen
   - Convert between timeline units.
   - Calculate the difference between two given timelines and display the value to the screen
2. Use **struct** to represent a fraction whose numerator and denominator are both integers. Write functions to complete the following requirements.
   - Construct a fraction from numerator and denominator entered by user and display the fraction to the screen.
   - Simplify the fraction, e.g. from 4/12 to 1/4
   - Perform arithmetic operations (i.e., +, -, *, /) between two given fractions and present the result as a simplest fraction

B. **FILE**

Read a **FILE** of *".txt"* extension whose filename is specified by user input. Write functions to complete the following requirements.

(a) Display the entire content of the file to the screen

(b) Count the number of words in the file

(c) Count the number of occurrences of a word, which is specified by user input, in the file

(d) Replace a word/phrase by another word/phrase and write the new text to a file of *".txt"* extension. Allow users to specify the location for saving.

(e) Write the first words of every sentences to a file has a file of *".txt"* extension. Allow users to specify the location for saving.

(f) Write all words that have more than 5 letters to a file of *".txt"* extension. Allow users to specify the location for saving.

2 HOMEWORK

1. Complete the requirements given in the File section above (In-Class/File) by using command line arguments. The following are commands for corresponding requirements.

   (a) `path/filename.exe  -printfile`

   (b) `path/filename.exe  -wordcount  -all`

   (c) `path/filename.exe  -wordcount  word_to_be_counted`

   (d) `path/filename.exe  -replace  word_to_be_replaced  word_replace  output_path/output.txt`

   (e) `path/filename.exe  -firstword  output_path/output.txt`

   (f) `path/filename.exe  -longword  output_path/output.txt`

2. Use command line argument to invoke a simple encryption/decryption program on a FILE of *".txt"* extension. The encryption method was introduced in the previous homework.

   (a) Encryption:

   - `path/filename.exe  -en  input_path/input.txt  key  output_path/output.txt`

   (b) Decryption:

   - `path/filename.exe  -de  input_path/input.txt  key  output_path/output.txt`

3 PREPARING YOUR SUBMISSION

Create a new folder and name it with your **Student ID**, e.g. 19127001. This folder includes

- **Code**: a sub-folder that contains your source code (*.cpp, *.h, etc.). Do not forget to delete all intermediate files.
- **Report** (if required): a sub-folder that contains your written report (*.pdf).