CMP2003 Data Structures and Algorithms (C++)
Term Project

-- Top 10 Frequent Words—

1. Project Definition

You are expected to write a c++ console application which reads files from Reuters-21578 documents collection appeared on the Reuters newswire in 1987 and find Top 10 frequent words used in the newswire articles. The Reuters-21578 collection is distributed in 22 files. Each of the first 21 files (reut2-000.sgm through reut2-020.sgm) contain 1000 documents, while the last (reut2-021.sgm) contains 578 documents.

Each article starts with an "open tag" of the form:

<REUTERS TOPICS=?? LEWISSPLIT=?? CGISPLIT=?? OLDID=?? NEWID=??>

where the ?? are filled in an appropriate fashion and ends with a "close tag" of the form:

</REUTERS>

Here is an example of these article entries in the file:

```
 <REUTERS ... >
 <DATE>26-FEB-1987 15:01:01.79</DATE>
<TOPICS><D>cocoa</D></TOPICS>
<PLACES><D>el-salvador</D><D>usa</D><D>uruguay</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN> ... </UNKNOWN>
<TEXT> ...
<TITLE>BAHIA COCOA REVIEW</TITLE>
<DATELINE>    SALVADOR, Feb 26 - </DATELINE>
  <BODY>Showers continued throughout
    the week in the Bahia cocoa zone, alleviating the drought since
    ...
    ...
    Brazilian Cocoa Trade Commission after
    carnival which ends midday on February 27.
   Reuter
  &#3;</BODY></TEXT>
</REUTERS>
```

Your program must be able to read words in articles in between <BODY> … </BODY> tags and insert each unique word into a suitable data structure.

Stopwords

A list of stopwords is supplied in stopwords.txt file. You should not count these words.

Definition of a word:

For simplicity assume that any contiguous block of alphabetic characters (letters from “a” to “z”, both upper and lower case) which includes at most one single quotation mark between these letters is a word. According to this definition the following sentence in a article:

" if we don't have local agreements settled by Thursday", has the words: “if”, “we”, “don’t”, “have”, “local”, “agreements”, “settled”, “by”, “Thursday”.

Main Requirements:

After reading and processing is over, your program must print “top 10” most frequent words used in these articles in **descending order**.

Additionally, the total time elapsed from the beginning of your code to the end of printing top 10 must be calculated and printed at the end of the execution.

Here is an example output:

```
<word1>      <word count>

<word2>      <word count>

<word3>      <word count>

<word4>      <word count>

<word5>      <word count>.

    .
    .
    .
    .
    .
    .
<word10>     <word count>

Total Elapsed Time: X seconds
```

Whole application can be implemented with console facilities (you do not need advanced GUI elements). The project consists of two parts.

A. Implementation of data structure.

This will be a proper **C++ class**. You must be able to create many instances of this class.
(Please use no third-party libraries and C++ STL, Boost etc.) However, you can use, **iostream**, **ctime**, **fstream**, **string** like IO and string classes.

B. The main program itself. In the main function, you must create a list of words.

2. Submission

You are expected to submit all source files(.cpp) and header files (.h, (if there are any)) of your project in a zip file with a presentation that is at most 15 minutes of video recording in **mp4** format. (You can find the requirements of video recording in **Evaluation s**ection.)

Only 1 group member must submit the project. Group members' names must be written in the submission.

Do not submit whole project solution or config files of your ide. And **do not submit *.sgm files with your work.!!!**

The project is at most **3 PERSONS** in size.

The deadline is set **January 22, 2021 11:59 pm.** Submit your files from **itslearning** system.

Late submissions will get lower grade by 25% for each day from submission deadline.

3. Cheating Policy.

You are not supposed to use each other's source code. Also please do not use source code from internet, another person or your book's/lab examples.

All the source codes will be filtered through a similarity analysis tool, which is known to be effective against many types of code copying and changing tricks. These projects will be graded as 0.

4. Evaluation

The most part (70%) of evaluation will depend on the implementation of data structure and correctness of your output.

We will sort all projects with respect to their running times, and you will get remaining of (30%) grade from this gradation.

Every group must prepare a video recording of their demonstration. And each group member must participate in the explanations given below:

- Run your code and show the output.
- Explain how you implemented the data structure on the code.
    - Explain classes and member functions (i.e. search or insert functions of the data structure.)

- Explain the part of the code that finds Top 10.

5. Bonuses

You can get bonuses for extra efforts:

* Good coding styles and OO programming skills

* Making a generic class for data structure.

* Or any other nice feature you can think of.

Please mention such extra efforts.

Notes:

1. Run your code in **“Release Mode”**, with an option **“full optimization”** to get the result quickly. (As a matter of fact, your code must run in Release Mode without crashing or any problem.)

2. You need to test your code in Visual Studio (any version is OK). All projects will be tested with Visual Studio. Please be sure that there is no compiler dependent problem occurs for your project.

3. A struct/class definition for “word” will be useful for storing the word and its count information together on the data structure you implement.