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

-- Top 10 Frequent Words—

1. Project Definition

You are expected to write a c++ console application that reads and counts unique words used in documents: articles, chapters, books about Applied sciences, Mathematics, Information science published from 1900 to 2021 and find Top 10 frequent words used in these documents.

ArticlesDataset.txt file contains all the metadata information of documents. **unigramCount** contains all unique words and their number of occurrences for each document. There are 1500 publications recorded in the txt file. Find total frequency of all the unigrams used in all publications and print top 10 frequent words in these documents. Here is an example entry for a document:

```
{"creator":["Romain Allais","Julie Gobert"],
"datePublished":"2018-05-30",
"docType":"article",
"doi":"10.1051\/mattech\/2018010",
"id":"ark:\/\/27927\/phz10hn2bh3",
"isPartOf":"Mat\u00e9riaux & Techniques",
"issueNumber":"5-6","language":["eng"],
"outputFormat":["unigram","bigram","trigram"],
"pageCount":7,
"pagination":"pp. null-null",
"provider":"portico",
"publicationYear":2018,
"publisher":"EDP Sciences",
"sequence":3.0,
"tdmCategory":["Applied sciences - Engineering"],
"title":"Environmental assessment of PSS",
"url":"http:\/\/doi.org\/10.1051\/mattech\/2018010",
"volumeNumber":"105",
"wordCount":4446,
"unigramCount":{"others":1,"air":1,"networks,":1,"conventional":1,"IEEE":1}
}
```

Your program must pull out the unigram counts for each document and store them in a suitable data structure.

Stop words

A list of stop words is supplied in stopwords.txt file to remove function words (e.g., the, and I, to, of, a) and other common words. 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 an article:

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

Also note that; to combine the counts of each unigram, you must lowercase all the characters in each string. That means the string "the" and "The" are same.

Main Requirements:

After reading and processing is over, your program must print “top 10” most frequent words used in these documents 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.
(**DO NOT** use third-party libraries and C++ STL, Boost etc.)
However, you can use, **iostream**, **ctime**, **fstream**, **string** like IO and string related classes.

B. Main program.

In the main function, you must create a list of words. And total time should be calculated from starting the preprocessing of data, to the end of Top 10 calculation.

2. Submission

You must submit:

- your code containing **ONLY** **source files(.cpp)** and **header files** (.h, (if there are any)) of your project in a **ZIP** file.

- **Report** (pdf format) and a **presentation** video that is at most 15 minutes of recording in **mp4** format. (You can find the requirements of report and video recording in **Evaluation** section.)

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

! **DO NOT SUBMIT** project configuration files (like .sln, etc.,) and **ArticlesDataset.txt** file.

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

The deadline is set **January 27, 2022 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's, or your book's/lab examples. If you use lab examples, you must improve the implementation.

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 (60%) of evaluation will depend on the implementation of data structure and correctness of the output.

10% of evaluation will depend on your Project Report of the performance evaluation of main data structures and algorithms used in the project.

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 search and insert functions of the data structure.
- Explain how your code finds Top 10 words and the reason of your search/sort algorithm choice.

Every group must prepare a project report with:

- Output of your code
- Main data structures and algorithms used in the project.
- Timing result

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 run on Visual Studio for evaluation. 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.