San José State University
Department of Computer Engineering

CMPE 180A
Data Structures and Algorithms in C++

Spring 2020
Instructor: Ron Mak

Assignment #11

Assigned:	Tuesday, November 3
Due:	Tuesday, November 10 at 5:30 PM
CodeCheck:	http://codecheck.it/files/201103082688s56dpaxo1k1ppt93446dlmt
Canvas:	Assignment 11: U.S. Constitution
Points:	100

U.S. Constitution

This assignment will give you practice with the built-in Standard Template Library (STL) vector, linked list, and map (hash table) by comparing their performances. Your program will read a text file and build three concordances, one with a sorted vector, one a sorted linked list, and one with a map.

A **concordance** is an alphabetical list of words from a document and their frequencies. Your input data will be a text file of the U.S. constitution and its amendments:
http://www.cs.sjsu.edu/~mak/CMPE180A/assignments/11/USConstitution.txt
This file is already uploaded to CodeCheck.

Your program should read each word of the text. If the word is not already in the concordances, enter the word with an initial count of 1 into both the vector, list, and map versions. If the word already exists in the concordance, increment the word's count by one in each concordance. The words in the concordance must be unique, and, of course, all three versions must end up with the same words and counts. Word comparisons should not be case-sensitive. Do not include numbers or punctuation marks.

Timings

Your program should keep track of how much time it takes to enter all the words into each concordance. For each word, compute the elapsed time only of the operation of either entering a new word into the concordance or incrementing the count of an existing word. Do not include the time to read the word from the input text file. Compare the total insertion times for the vector vs. the list vs. the map. The timings should be in microseconds (usec).

After building the three concordances, your program should compare the total time it takes to do 10,000 random word searches in each concordance. Since the vector-based concordance will be sorted, use a **binary search**.

Verification

Your program should use a list of words, both in and not in the concordance, to make (untimed) spot checks of the completed concordances to make sure they all agree on the frequency counts of those words.

Use iterators to iterate over the completed vector, list, and the map versions of the concordance in parallel to ensure that they contain the same data (words and counts) in the same order. Note that the elements of an STL map are always sorted by their keys.

Sample output

Your output should be similar to the example. Since there are timings, CodeCheck will not compare your output. Ignore the score 0. However, you should visually check that your word frequencies match the CodeCheck results.

What to submit

Submit the signed zip file into Canvas: **Assignment #11: U.S Constitution**. Also submit a text file containing your program's output.

You can submit as many times as necessary to get satisfactory results, and the number of submissions will not affect your score. When you're done with your program, click the "Download" link at the very bottom of the Report screen to download the signed zip file of your solution.

Rubrics

Criteria	Max points
Statistics (should be the same as the sample output) • Total words • Distinct words (vector, list, and map sizes)	20 • 10 • 10
Word insertions • Vector timing • List timing • Map timing	30 • 10 • 10 • 10
Checks • Spot checks (same frequency counts as the sample output) • Matching vector, list, and map concordances	20 • 10 • 10
Word searches • Vector timing (with binary search) • List timing • Map timing	30 • 10 • 10 • 10

```
Timed insertions
----------------
      Lines:    865
 Characters: 43,976
      Words:  7,541

Vector size: 1,138
  List size: 1,138
   Map size: 1,138

Vector total insertion time:  19,157 usec
  List total insertion time: 480,094 usec
   Map total insertion time:  11,888 usec

Spot checks of word counts
--------------------------
      amendment: vector: 35   list: 35  map: 35
        article: vector: 28   list: 28  map: 28
         ballot: vector:  5   list:  5  map:  5
       citizens: vector: 18   list: 18  map: 18
       congress: vector: 60   list: 60  map: 60
   constitution: vector: 25   list: 25  map: 25
      democracy: vector:  0   list:  0  map:  0   !!!
       electors: vector: 16   list: 16  map: 16
     government: vector:  8   list:  8  map:  8
            law: vector: 39   list: 39  map: 39
    legislature: vector: 13   list: 13  map: 13
         people: vector:  9   list:  9  map:  9
       president: vector:121   list:121  map:121
representatives: vector: 29   list: 29  map: 29
          right: vector: 14   list: 14  map: 14
          trust: vector:  4   list:  4  map:  4
         united: vector: 85   list: 85  map: 85
           vice: vector: 36   list: 36  map: 36
           vote: vector: 16   list: 16  map: 16

Checking concordance versions
-----------------------------
All match!

Timed searches (10,000 searches)
--------------
Vector total search time:  13,210 usec
  List total search time: 493,060 usec
   Map total search time:  11,514 usec

Done!
```

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.