CISC 3130 Ari Mermelstein

Homework #3 : Due March 12

In this homework, you will be implementing a Lexicon class. This Lexicon class will maintain a list of words, and the user will be able to query the Lexicon as to whether or not a word is in the Lexicon, the user will be able to add words to the Lexicon, or to modify a word in the Lexicon.

Constructors:

Lexicon();

A constructor that allows the user to create an empty Lexicon:

Lexicon(const string& fileName);

A constructor that opens the file associated with *fileName*, and stores all of the words in the file in the Lexicon. If the file doesn't exist, an exception should be thrown.

(N.B. The ifstream constructor takes a C-style string as a parameter to open the file and not a C++ string object. To get the C-string equivalent of any given C++ string, use the member function c_str(). For example, to get the C-string equivalent of *fileName*, call fileName.*c_str().)*

Operations:

bool containsWord(const string& word);

returns true if *word* is in the Lexicon, and false otherwise.

list<string> startWith(char ch);

returns a list a of words that start with a particular letter, *ch*.

list<string> startWith(const string& prefix);

returns a list of words that start with *prefix*.

list<string> endWith(char ch);

returns a list of words that end with a particular character, *ch*.

list<string> endWith(const string& suffix);

returns a list of words that end with *suffix*.

void addWord(const string& str);

Adds *str* to the Lexicon if it is not there already. If it is there already, do nothing.

void updateWord(const string& target, const string& replacement);

Changes the spelling of *target* to become *replacement*.

Operators:

friend bool operator==(const Lexicon& d1, const Lexicon& d2);

returns true iff d1 and d2 contain the same words, not necessarily in the same order.

(N.B. be careful here. You can not just assume that you can compare the lexicons word by word, since they may not be stored in the same order.)

friend operator <<(ostream& os, const Lexicon& d);

Outputs the content of the Lexicon (in any order).

Please submit Lexicon.h, Lexicon.cpp, and useLexicon.cpp files. Also, please use comments to document your code. These homeworks will become more and more complicated, and if I can't figure out what your intuition is in solving these problems, I can't give credit.

I have provided 2 text files with the same lexicon, but I ran a program that has randomly permuted the words such that they are no longer listed in sorted order. When these files have been stored into Lexicon objects, your == function should return true.

Source: https://github.com/dwyl/english-words/blob/master/words.txt