Comp625: Homework 3 due October 14th 2022[1]

Self-referential Strings

Synopsis

For your third homework you will write exhaustive search to find all possible self-referential strings given a dictionary of words.

Task

Your program can assume proper input and will not need to worry about error handling (please avoid adding code that does). Your program should store a list of characters (possible letters) and a list of strings (your sentence of referential strings) and then proceed to exhaustively step through the list of strings to derive every possible combination that incorporates your dictionary words. Dictionary words will be: one, two, three, four, five, six, seven, eight, nine, ten. There will be two special strings: “?” and “#” in each sentence. The first will be substituted by each character and the second should be replaced by all possible dictionary words. The aim is to have the sentence properly report the number of characters in it.

Here’s a run of your program:

```
C:\> Homework3.exe
Name of sentence file to read in: sentences.txt
File successfully read in!
Sentence 1:
Letter a occurs one times
Letter b occurs one times
Letter c occurs three times
Letter d occurs one times
Letter e occurs four times
Letter e occurs five times
Sentence 2:
The letter s appears in this sentence five times
The letter s appears in this sentence six times
The letter t appears in this sentence seven times
The letter t appears in this sentence eight times
The letter r appears in this sentence four times
ERROR: no match for h
C:\>
```

An example of file `sentence.txt` looks as follows:

```
2
5 a b c d e
5 Letter ? occurs # times
4 s t r h
9 The letter ? appears in this sentence # times
```

[1] Due promptly at 11:59pm.

Homework3.cpp looks like this:

```
// This is the main code. Do not modify.
#include "Sentence.h"

int main()
{
    Sentence *self = new Sentence();
    self->run();
    delete self;
}
```

Notes:

Your searches for each character in the string should ***not be case sensitive***. Look both in C library's <string.h> and C++ class's <string> include files to find appropriate string functions to help.

Note: it's helpful to use Google to search for say "C++ string methods" and "C string functions" (my recommendation).

You must create two arrays for your data structures, one that stores separate letters (can be either of type character or type string) and one that stores separate words (an array of strings). You will then need to read in the letters and the words in the string and use it to create all possible sentences using your internal dictionary. Note you do not have to read in the entire file at once, instead reading in one series of letters and words and computing the letter count for each sentence generated for that pairing.

You will need to create at least one other class with several methods in them. You will not need to use inheritance. You will be graded not only on correctness but also on how well your class is designed, and use proper coding techniques (i.e know when a for loop is better than a while loop, etc…)

Please follow user interface exactly as it appears in this document (otherwise you'll crash the grading program and that will cost you some valuable points).

Submission:

Put source files (excluding Homework3.cpp) into folder named ***First.Last.Homework3***, then **Zip** and attach to email with subject: **COMP625 Homework3**