C.I.S. 1.5 Brooklyn College Professor Langsam

Assignment #5

Write a C++ program that encodes English-language phrases into pig-Latin. Pig-Latin is a form of coded language often used for amusement. According to Wikipedia:

Pig-Latin is a language game primarily used in English. An alternative British name for Pig-Latin is **backslang** (in Britain this term more often applies to the type of backslang used by the criminals of 19th century London and used as a playground game today, which was based on turning words backwards), or Butcher's Backslang which was common in English Butcher's shops at least until World War II. Prior to this, Benjamin Franklin was known to use a version of Pig-Latin in some publications. Pig-Latin is usually used by children for amusement or to converse in (perceived) privacy from adults or other children. Conversely, adults sometimes use it to discuss sensitive topics they do not want very young children to overhear.

Many variations exist in the methods used to form pig-Latin phrases. For simplicity, use the following algorithm:

To form a pig-Latin phrase from an English-language phrase, tokenize the phrase into words. Place the first letter of the English word at the end of the English word and add the letters "ay." Thus the word "jump" becomes "umpjay," the word "the" becomes "hetay" and the word "computer" becomes "omputercay." Blanks and punctuation marks between words remain unchanged. If a word in the English phrase was capitalized your will have to change the case of the pig-Latin phrase accordingly. Thus the word "Professor" would become "Rofessorpay."

Use the text file "GettyburgAddress.txt" found at

http://eilat.sci.brooklyn.cuny.edu/cis1_5/CISClassPage.htm

At the end of your program print the following statistics:

- 1. the number of words in the text.
- 2. the number of letter in the text.
- 3. the number of all characters in the text (including punctuation and whitespace.)

Suggestions:

- While developing and testing your program use a smaller datafile than the one specified above.
- Scan and process the text one character at a time, looking for white-spaces, using the techniques we developed in class.
- You may modify the character processing routines we developed in class.

Be sure to use the structured programming techniques we have learned in class. Do not use external variables (with the exception of file variables), unless you are prepared to justify their necessity. Use meaningful variables and comment each function.