CPSC 2430-02 Fall 2017 Programming Assignment #5
Tuesday, November 21, 2017 at Midnight

P5 exercises your understanding of Hash functions, Hash tables and collision resolution

Write a program that uses **hashing** for the following problem. Given a natural language text, generate a table of distinct words with the number of occurrences of each word in the text. A word is defined as a series of **alphabetic characters of length > 5**. Note that punctuation and spaces can both (separately or together) delineate a word. Capitalized versions of a word are considered the same as the lowercase version, so feel free to change all words to either upper- or lowercase.

The book Ulysses by James Joyce is a story about a day in the life of several Irish citizens. It can be found at http://www.gutenberg.org/ebooks/4300. The text version is the input for your program.

Your resulting program will be an interactive one allowing the user to type in a word (that may or may not appear in the table) and receive the number of occurrences of that word in the text until the user wishes to quit.

The challenge: Find the combination of *hash function and hash table size, using linked lists as your collision resolution* that will minimize the collisions, while maintaining a "reasonable" table size. A collision is defined as an attempt to store a new word in a location already occupied by another word. Consequently, it is possible for a new word to collide more than once before it is finally stored.

Program description: Your program should hash the words from the file and then print (neatly formatted) the number of collisions, number of unique words and total number of words to the standard output (screen) before allowing the user to input a word to see the number of times it appeared. The user should be able to input as many words as desired before choosing to quit.

Suggestions & Notes:

- Reading in one word at a time from the file, create a function that pre-processes a word, changing capital letters to lowercase and removing any quotation marks or other unnecessary punctuation that may have been read from the file. Then send the pre-processed string into your hash function.
- When counting collisions, do not count additional instances of a word. So you will want to keep a "temporary" collision count for a word until you know if the word is already in the table.
- Note that the table size must be "reasonable" and depends upon the number of entries expected and the collision resolution used.

- It is suggested that you get your program up and running by using one of the hash functions discussed in class. Make sure you are pre-processing your words correctly, etc. before attempting to minimize the collisions.

Call your file “p5.cpp”. Submit your project by typing the following command from the prompt in the directory where the files are located:
/home/fac/sreeder/submit/cpsc2430/p5_runme