COSC 2436 lab2: Text Processing and Sorting in Linked Lists

Create a C++ program to read in all the words from a text file and group the words together. Your program should read in each word and remove any extra punctuation. It should then store the word in a linked list, with a count of the number of times that word has appeared. Once all the words have been entered, it should sort the entries from most common to least common. Your program should write the results to a file showing in the count of each word, a space, and the word.

1. Input files
   - The input files contain text, including spaces and punctuation. First read in the words and remove all unnecessary punctuation. You should keep only the apostrophe **(')** and hyphen **(-)** as they can be part of valid words. For example: **you're** is a word and is distinct from **you**. It should **not be** processed as two "words": **you** and **re**.
   - The words should be stored in a Linked-List along with a count of the number of times the word occurs.
   - For simplicity, convert all words to upper-case to prevent miscounting of words at the start of a sentence. The words **"They"** and **"they"** are the same and should be counted together as **"THEY"**. The words **"let's"** and **"lets"** are different.
   - The linked list should then be sorted in **decreasing** order by the number of occurrences of each word.
   - If two words have the same number of occurrences, they should be arranged in alphabetical order (A-Z).

2. Output files
   - Output the count, followed by a single space, and then the word for all the entries in the linked list.
   - The words should be output in all-caps and include only valid punctuation.
   - Output the results in decreasing order of their occurrence in the file. If multiple words have the same number of occurrences, save them in alphabetical order.

3. Example

input1.txt	output1.txt
Baby Shark, doo-doo, doo-doo Baby Shark, doo-doo, doo-doo Baby Shark, doo-doo, doo-doo Baby Shark Mommy Shark, doo-doo, doo-doo Mommy Shark, doo-doo, doo-doo Mommy Shark, doo-doo, doo-doo Mommy Shark Daddy Shark, doo-doo, doo-doo Daddy Shark, doo-doo, doo-doo Daddy Shark, doo-doo, doo-doo Daddy Shark Grandma Shark, doo-doo, doo-doo Grandma Shark, doo-doo, doo-doo Grandma Shark, doo-doo, doo-doo Grandma Shark Grandpa Shark, doo-doo, doo-doo Grandpa Shark, doo-doo, doo-doo Grandpa Shark, doo-doo, doo-doo Grandpa Shark Let's go hunt, doo-doo, doo-doo Let's go hunt, doo-doo, doo-doo Let's go hunt, doo-doo, doo-doo Let's go hunt Run away, doo-doo, doo-doo Run away, doo-doo, doo-doo Run away, doo-doo, doo-doo Run away (ah!) Safe at last, doo-doo, doo-doo Safe at last, doo-doo, doo-doo Safe at last, doo-doo, doo-doo Safe at last (phew) It's the end, doo-doo, doo-doo It's the end, doo-doo, doo-doo It's the end, doo-doo, doo-doo It's the end	54 DOO-DOO 20 SHARK 4 AT 4 AWAY 4 BABY 4 DADDY 4 END 4 GO 4 GRANDMA 4 GRANDPA 4 HUNT 4 IT'S 4 LAST 4 LET'S 4 MOMMY 4 RUN 4 SAFE 4 THE 1 AH 1 PHEW

4. Reminder
   - Turn in your lab assignment to our Linux server, follow the link here for more instructions.
   - Make sure to only have **one (1)** .cpp file with the main() function in your working directory, otherwise your program will fail the grading script.
     - Create a folder under your root directory, name the folder *lab2* (case sensitive), copy all your .cpp and .h files to the folder (ArgumentManager.h is also needed)
     - Only include the necessary files (.cpp and .h files) in your working directory in your final submission
     - To test your program, copy the input files into the server and run your program. After verifying that they pass, delete the .txt files.

Please reach out to myself or the TAs for any clarifications or typos.