Juliana McCausland
Ling 473
Project 1

Results:

the	4398064
a	2032214
to	1893205
of	1888409
and	1759680

…and so on

Project Description:

This project ended up requiring less code than I anticipated. The suggestions in the spec were helpful and led to a straightforward approach. The program iterates through each file in the directory, calling the file_cleanup function, which performs the crucial task of cleaning the text so that what remains are just the words (or what we are defining as words in this context). I encountered a slight roadblock when I initially attempted to use a single line of code that included all regex patterns in one (separated by '|'), and got unexpected results. Breaking up the patterns into separate re.sub functions somehow resolved this problem, although I'm not sure why. After all the re.sub commands are completed, the clean text is split into a list, and the function returns this value.

The program iterates through the list, adding each word to a dictionary, and incrementing the count value each time a word is encountered beyond its first encounter. The key-value pairs from the dictionary are appended to a final list, which is then sorted and printed according to formatting instructions.

Overall, the biggest challenge was figuring out the regex portion. I struggled with implementing the correct patterns without making extra, unwanted changes to data that I wanted to preserve. I initially planned to use NLTK, but realized I that was not necessary. I was also worried that passing the text between strings and lists and dictionaries would consume significant running time, but it ended up causing less havoc than I expected. I am sure there is a more efficient way of coding this kind of task, though!