# CST126

## Lab 3 – Author, Author

In this lab you will create a program that counts the words in a file.

### Overview

In 1787 and 1788 a series of 85 political essays, called The Federalist Papers were published under the pseudonym “Publius.” Authorship of these papers is now attributed to Alexander Hamilton, James Madison and John Jay.[[1]](#footnote-1) To help determine who wrote the essays, historians compared the choice of words in each essay with the choice of words in known publications of the suspected authors. In this program you will aid this historical quest by creating a program that counts the number of words in a file.

### Input

I have provided three test federalist papers which I downloaded and updated from

<http://memory.loc.gov/const/fed/fedpapers.html> [[2]](#footnote-2) . You can assume the following about these test data files:

1. The first line will have the number of words in the file.
2. Some words may have the first letter capitalized. You should lowercase the first character. For example the word “By” and the word “by” count as 2 occurrences of the same word.
3. Some words may be preceded by punctuation. (for example “) You should strip any proceeding punctuation before counting a word.
4. Some words may be followed by punctuation. (For example ! or , ) You should strip any punctuation at the end of the word before counting the word.
5. Words that have a hyphen in them are considered a single word, for example well-grounded is one word.

You will want a smaller test file prior to using the larger federalist files. I included a file “little.txt” to help you with this.

You will ask the user to enter the file name.

### Logic

1. Use a class to store your words and their counts.
2. You are required to use a dynamically allocated array to store the list of words. In other words, a dynamically allocated array of classes. You are required to delete this array before exiting the program.
3. Your program must sort your list by descending word frequency. Don’t worry about doing a secondary sort. Any order is fine for words with the same frequency.

### Output

Your program should output the file name and then a nicely formatted lists of the top 10 most frequent words and their frequency. The words and their frequency should be sorted as described in the logic section. I have provided sample output for fed1.txt (fed1.out). I will test with all three files (fed1.out,fed2.out and fed6.out).

Please enter the input file name including extension : fed1.txt

the 125

of 105

to 69

and 38

be 33

that 26

a 24

in 24

will 23

which 18

Press any key to continue . . .

**Hints**

1. You can allocate your array immediately after reading the first line in the input file.
2. In lab I will show you my version of an array sorting program that sorts in descending order.
3. I created a routine “larger” that compares the two structures. It returns true if word1 is bigger than word2 based on the sorting criteria in the logic section. Here is the prototype:

bool larger(word word1, word word2);

During lab I will show you a similar routine for a different structure.

1. The idea of testing each function as you create it is very important in this lab. If you have a problem with your sort it will be much easier to locate if you test each component routine separately
2. The routine ispunct will tell you if a specific character is punctuation.

1. <http://www.infoplease.com/ce6/history/A0818386.html>, The Columbia Electronic Encyclopedia Copyright © 1994, 2000, Columbia University Press. Licensed from Columbia University Press, viewed 8 May 2018. [↑](#footnote-ref-1)
2. http://memory.loc.gov/const/fed/fedpapers.html Last Update Wed Dec 25 1996, viewed 8 May 2018. [↑](#footnote-ref-2)