4/1/2014

Allison Channic

Lab time: Thurs. 12:00

Lily Lu

Lab time: Thurs. 2:00

**Program 4: Analyze this**

Requirements Summary

We believe that we have met all of the requirements as stated in the program description, however, our code probably could have been more efficient. For example, we hard coded all of the file names for each speech, each president’s name, and the year each speech was given as opposed to reading them in from a file. For the two methods that counted first-person pronouns, we had the same code multiple times just for different words and their occurrences. We did not do any of the extra credit.

Table of Results

President Year WC SC PC AWL ASL FPSP FPPP SP/PP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.) George Washington 1789 1424 23 8 4.96 61.9 52 3 17.33

2.) George Washington 1793 132 4 3 4.92 33.0 8 0 N/A

3.) John Adams 1797 2308 37 14 4.89 62.4 32 13 2.46

4.) Thomas Jefferson 1801 1719 41 7 4.76 41.9 32 44 0.73

5.) Thomas Jefferson 1805 2150 45 15 4.89 47.8 33 46 0.72

6.) James Madison 1809 1172 21 7 4.89 55.8 32 14 2.29

7.) James Madison 1813 1200 33 11 4.87 36.4 9 27 0.33

8.) James Monroe 1817 3361 123 21 4.81 27.3 37 104 0.36

9.) James Monroe 1821 4448 133 28 4.80 33.4 48 82 0.59

10.) John Quincy Adams 1825 2911 76 10 5.01 38.3 35 48 0.73

11.) Andrew Jackson 1829 1117 25 13 5.01 44.7 32 22 1.45

12.) Andrew Jackson 1833 1169 30 10 4.94 39.0 25 28 0.89

13.) Martin Van Buren 1837 3831 98 20 5.02 39.1 85 101 0.84

14.) William Henry Harrison 1841 8457 219 25 4.79 38.6 89 86 1.03

15.) James Polk 1845 4789 153 31 4.90 31.3 48 113 0.42

16.) Zachary Taylor 1849 1082 22 10 5.00 49.2 33 22 1.50

17.) Franklin Pierce 1853 3332 104 15 4.92 32.0 59 58 1.02

18.) James Buchanan 1857 2815 89 22 4.87 31.6 23 49 0.47

19.) Abraham Lincoln 1861 3608 135 39 4.70 26.7 47 23 2.04

20.) Abraham Lincoln 1865 697 26 5 4.52 26.8 2 11 0.18

21.) Ulysses S. Grant 1869 1122 40 15 4.68 28.1 30 17 1.76

22.) Ulysses S. Grant 1873 1326 43 17 4.70 30.8 48 16 3.00

23.) Rutherford B. Hayes 1877 2472 59 28 4.92 41.9 35 27 1.30

24.) James Garfield 1881 2954 111 39 4.90 26.6 16 49 0.33

25.) Grover Cleveland 1885 1673 44 19 4.96 38.0 11 34 0.32

26.) Benjamin Harrison 1889 4362 157 36 4.90 27.8 20 109 0.18

27.) Grover Cleveland 1893 1992 58 27 5.11 34.3 31 74 0.42

28.) William McKinley 1897 3966 130 19 4.85 30.5 38 88 0.43

29.) William McKinley 1901 2210 100 13 4.95 22.1 19 64 0.30

30.) Theodore Roosevelt 1905 985 33 5 4.55 29.8 0 69 0.00

31.) William Taft 1909 5397 160 44 4.87 33.7 50 72 0.69

32.) Woodrow Wilson 1913 1696 68 11 4.52 24.9 7 71 0.10

33.) Woodrow Wilson 1917 1513 59 17 4.43 25.6 14 91 0.15

34.) Warren Harding 1921 3305 148 33 4.99 22.3 20 157 0.13

35.) Calvin Coolidge 1925 4032 196 24 4.83 20.6 9 152 0.06

36.) Herbert Hoover 1929 3816 169 51 5.05 22.6 25 120 0.21

37.) Franklin D. Roosevelt 1933 1856 86 25 4.75 21.6 29 60 0.48

38.) Franklin D. Roosevelt 1937 1781 96 37 4.80 18.6 12 85 0.14

39.) Franklin D. Roosevelt 1941 1304 74 38 4.60 17.6 2 46 0.04

40.) Franklin D. Roosevelt 1945 533 29 17 4.43 18.4 6 33 0.18

41.) Harry S. Truman 1949 2289 120 72 4.84 19.1 8 82 0.10

42.) Dwight D. Eisenhower 1953 2478 128 53 4.49 19.4 8 132 0.06

43.) Dwight D. Eisenhower 1957 1678 96 42 4.28 17.5 4 76 0.05

44.) John F. Kennedy 1961 1391 57 28 4.32 24.4 6 60 0.10

45.) Lyndon B. Johnson 1965 1514 95 35 4.27 15.9 22 80 0.28

46.) Richard Nixon 1969 2144 107 73 4.24 20.0 20 125 0.16

47.) Richard Nixon 1973 1855 74 52 4.34 25.1 18 100 0.18

48.) Jimmy Carter 1977 1239 54 32 4.41 22.9 17 80 0.21

49.) Ronald Reagan 1981 2427 135 37 4.49 18.0 26 131 0.20

50.) Ronald Reagan 1985 2569 127 42 4.51 20.2 20 140 0.14

51.) George Bush 1989 2332 154 28 4.19 15.1 35 113 0.31

52.) Bill Clinton 1993 1599 92 17 4.53 17.4 13 120 0.11

53.) Bill Clinton 1997 2155 110 24 4.49 19.6 6 129 0.05

54.) George W. Bush 2001 1590 96 30 4.52 16.6 12 103 0.12

55.) George W. Bush 2005 2070 100 28 4.62 20.7 9 89 0.10

56.) Barack Obama 2009 2405 122 29 4.42 19.7 4 153 0.03

57.) Barack Obama 2013 2098 91 30 4.55 23.1 6 158 0.04

Analysis

Word count, sentence count, and paragraph count appear to have no patterns until the last few (6-7) speeches. At that point, word count is approximately 200, sentence count is approximately 100, and paragraph count is around 20 – 25. This could be attributed to the trend of presidents not writing speeches themselves and/or the multitude of data available on previous inaugural addresses due to technological advances and an increase in knowledge of history, linguistics, and rhetoric.

Average word length tended to remain constant for all of the speeches, floating at around 5. Average sentence length, however, seems to have generally decreased with time. Reasons may include decrease in literacy, education, or stylistic purposes.

The number of first-person singular pronouns varied greatly for each speech, yet there has been a drastic increase in the number of first-person plural pronouns used since President Dwight D. Eisenhower took office. Perhaps this started with cold-war, anti-communist sentiment that promoted a sense of nationalism (“our” country as opposed to “my” country) and was a trend that just stuck around for whatever reason.

Likewise, the ratio of first-person singular pronouns to first-person plural pronouns also decreased with time. Noticeably, from the time Andrew Jackson took office until after Rutherford B. Hayes (1829 – 1877), the ratio was significantly higher. The most probable cause for such an increase was either a political or socio-economic trend or occurrence that prompted these respective presidents to utilize singular pronouns in their speech.

Based on these results, it appears that there are specific time periods in which certain linguistic properties and trends can be found. To further understand this data, perhaps someone more knowledgeable in U.S. History or linguistics should analyze these findings in order to provide better context on the both the social and political historical events that took place during this time and how language would possibly be affected by such. Equally important is each president’s political party, which was not included in this project. Perhaps certain linguistic characteristics were found for presidents of a certain political party. Again, someone with a better understanding of political history might be better to comment on such trends, in order to accurately account for the shift in political party views over time (i.e. Abraham Lincoln was a republican, yet his political views would certainly not be comparable to someone like Ronald Reagan or George W. Bush).

Code

(Copied and pasted from Allison’s computer)

/\* ----------------------------------------------------------------------------

\* Analyze This

\* CS 141 Spring 2014

\*

\* Allison Channic

\* Lab Time: Thurs. 12:00

\* System: Windows 8, written in code writer, compiled with visual studio

\*

\* Lily Lu

\* Lab Time: Thurs. 2:00

\* System: Windows 8, DevC++

\* ----------------------------------------------------------------------------

\*

\* Analyze This

\*

\* The program takes the inaugural addresses from each president of the United

\* States and calculates the number of words, sentences, paragraphs, average

\* sentence length, average word length, the number of first person singular

\* pronouns, the number of first person plural pronouns, and the ratio of first

\* person singular pronouns to first person plural pronouns. Each inaugural

\* address is in its own .txt file in a directory called "datafiles". The

\* program opens each file, sets the text to a dynamically allocated array,

\* and then analyzes the array accordingly. The data from each inaugural

\* address is then printed in a table format.

\*

\* Running the program looks like this:

\*

President Year WC SC PC ASL AWL FPSP FPPP SP/PP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.) George Washington 1789 1424 23 7 62.04 4.96 57 3 19.00

2.) George Washington 1793 132 5 3 33.75 4.89 9 0 N/A

3.) John Adams 1797 2308 37 14 62.68 4.89 34 14 2.43

4.) Thomas Jefferson 1801 1719 41 7 41.88 4.76 34 44 0.77

5.) Thomas Jefferson 1805 2150 45 15 47.93 4.87 37 46 0.80

6.) James Madison 1809 1172 21 7 55.86 4.88 32 14 2.29

7.) James Madison 1813 1200 33 11 36.67 4.83 10 27 0.37

8.) James Monroe 1817 3361 123 21 27.37 4.80 38 105 0.36

9.) James Monroe 1821 4448 133 28 33.54 4.78 50 85 0.59

10.) John Quincy Adams 1825 2911 76 10 38.25 5.01 35 49 0.71

11.) Andrew Jackson 1829 1117 25 13 44.96 4.98 32 22 1.45

12.) Andrew Jackson 1833 1171 30 10 39.03 4.93 28 28 1.00

13.) Martin van Buren 1837 3831 98 20 39.04 5.02 89 104 0.86

14.) William Henry Harrison 1841 8457 219 25 38.47 4.81 93 87 1.07

15.) James K. Polk 1845 4789 153 31 31.33 4.90 51 116 0.44

16.) Zachary Taylor 1849 1082 22 10 49.41 4.98 33 22 1.50

17.) Franklin Pierce 1853 3332 104 15 32.01 4.93 61 59 1.03

18.) James Buchanan 1857 2815 89 22 31.69 4.86 25 51 0.49

19.) Abraham Lincoln 1861 3608 135 39 26.87 4.67 58 23 2.52

20.) Abraham Lincoln 1865 697 26 5 26.85 4.50 2 11 0.18

21.) Ulysses S. Grant 1869 1122 40 15 28.02 4.68 31 17 1.82

22.) Ulysses S. Grant 1873 1326 43 17 31.05 4.67 51 16 3.19

23.) Rutherford B. Hayes 1877 2472 59 28 41.88 4.93 38 28 1.36

24.) James Garfield 1881 2954 111 39 26.78 4.87 19 52 0.37

25.) Grover Cleveland 1885 1673 44 19 38.11 4.95 11 34 0.32

26.) Benjamin Harrison 1889 4362 157 36 27.93 4.88 24 115 0.21

27.) Grover Cleveland 1893 1992 58 27 34.71 5.06 35 75 0.47

28.) William McKinley 1897 3966 130 19 30.44 4.86 39 90 0.43

29.) William McKinley 1901 2210 100 13 22.15 4.93 21 67 0.31

30.) Theodore Roosevelt 1905 985 33 5 29.79 4.56 1 73 0.01

31.) William Taft 1909 5397 160 44 33.92 4.84 55 76 0.72

32.) Woodrow Wilson 1913 1696 68 11 24.99 4.51 8 75 0.11

33.) Woodrow Wilson 1917 1513 59 17 25.80 4.40 17 94 0.18

34.) Warren Harding 1921 3305 148 33 22.45 4.97 26 164 0.16

35.) Calvin Coolidge 1925 4032 196 24 20.69 4.80 10 155 0.06

36.) Herbert Hoover 1929 3816 169 52 22.76 5.01 29 125 0.23

37.) Franklin D. Roosevelt 1933 1856 86 25 21.86 4.69 31 63 0.49

38.) Franklin D. Roosevelt 1937 1781 96 37 18.77 4.74 18 88 0.20

39.) Franklin D. Roosevelt 1941 1304 74 34 18.05 4.49 2 53 0.04

40.) Franklin D. Roosevelt 1945 533 29 17 18.79 4.32 8 42 0.19

41.) Harry S. Truman 1949 2289 120 73 18.91 4.89 12 98 0.12

42.) Dwight D. Eisenhower 1953 2478 128 53 19.11 4.55 12 138 0.09

43.) Dwight D. Eisenhower 1957 1678 96 42 17.04 4.39 5 91 0.05

44.) John F. Kennedy 1961 1391 57 28 23.53 4.48 8 63 0.13

45.) Lyndon B. Johnson 1965 1514 95 35 15.77 4.32 24 83 0.29

46.) Richard Nixon 1969 2144 107 71 19.65 4.32 29 137 0.21

47.) Richard Nixon 1973 1855 74 52 24.77 4.39 20 107 0.19

48.) Jimmy Carter 1977 1239 54 27 22.43 4.52 18 88 0.20

49.) Ronald Reagan 1981 2427 135 36 17.94 4.50 28 138 0.20

50.) Ronald Reagan 1985 2569 128 41 20.00 4.53 25 150 0.17

51.) George Bush 1989 2332 154 28 15.06 4.22 39 117 0.33

52.) Bill Clinton 1993 1599 92 17 17.37 4.53 15 124 0.12

53.) Bill Clinton 1997 2155 110 24 19.57 4.50 8 134 0.06

54.) George W. Bush 2001 1590 96 30 16.57 4.52 15 110 0.14

55.) George W. Bush 2005 2070 100 28 20.69 4.63 10 92 0.11

56.) Barack Obama 2009 2405 122 29 19.60 4.45 5 156 0.03

57.) Barack Obama 2013 2098 91 29 22.96 4.57 7 168 0.04

\*

\*/

//Include necessary C/C++ libraries

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

/\* Define a speech struct

\* A speech has four elements: the year it was given, an array containing

\* the text characters of the speech, the file name associated with that speech

\* in the directory, and the name of the president who gave the speech.

\*/

typedef struct Speech\_struct

{

int year; //year speech was given

char \*speechArray; // pointer to speech file

char speechFileName[100]; //name of speech file

char presidentName[100]; //name of president who gave speech

}Speech;

/\* Function to get the number of alphabetic characters in a speech

\* Takes a pointer to a char array as its parameter

\* Uses a pointer variable to go through the array and sees if each character

\* is a letter of the alphabet

\* Returns the total number of alphabetic characters as an int

\*/

int getNumAlphabetic (char \*&speechArray)

{

//Pointer to go through the array

char \*arrayPointer = speechArray;

//Variable to represent the number of alphabetic characters

int numAlphabetic = 0;

//Variable to represent each character in the array

int i =0;

//Check to see if each character is from the alphabet and increment the

//number of alphabetic characters and the array pointer accordingly

while(speechArray[i] != NULL)

{

if(isalpha(speechArray[i]))

{

numAlphabetic++;

arrayPointer++;

}

i++;

}

//Return the total number of alphabetic characters

return numAlphabetic;

}

/\* Method to get the number of words

\* Takes a pointer to an array containing the text from a speech file

\* as its parameter

\* Looks through the array and counts the number of spaces and hyphens

\* Does this using a pointer variable to the speech array that advances

\* through the text after each occurrence of a respective character

\* For each space and hyphen, a variable "numWords" is increased by one

\* to account for a space being at the end of every word or a hyphen separating

\* two words

\* Returns the number of words as an integer

\*/

int getNumWords(char \*&speechArray)

{

//Pointer variable used to advance through the speech array

char \*arrayPointer = speechArray;

//Variable to accumulate the number of words

int numWords = 0;

//Count the number of spaces

while(arrayPointer != NULL)

{

arrayPointer = strchr(arrayPointer, ' ');

if( arrayPointer != NULL)

{

numWords++;

arrayPointer++;

}

}

//Reset the array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Count the number of hyphens

while(arrayPointer != NULL)

{

arrayPointer = strchr(arrayPointer, '-');

if( arrayPointer != NULL)

{

numWords++;

arrayPointer++;

}

}

//Return the total number of spaces and hyphens

return numWords;

}

/\* Method to get the number of sentences

\* Takes a pointer to an array containing the text from a speech file

\* as its parameter

\* For this program, a sentence is defined as any fragment of text that ends in

\* a '.', '?', or '!' character that isn't an ellipses

\* Counts the number of '.', '?', and '!' characters that aren't associated

\* with ellipses by looking through the speech array, increasing a variable

\* "numSentences" when a punctuation character is found, then advances to the

\* next character using an array pointer to the speech array

\* Returns the total number of sentences as an integer

\*/

int getNumSentences(char \*&speechArray)

{

//Pointer variable used to advance through the speech array

char \*arrayPointer = speechArray;

//Variable for the total number of sentences

int numSentences = 0;

//Check for sentences ending with an exclamation mark

while(arrayPointer != NULL)

{

arrayPointer = strchr(arrayPointer, '!');

if( arrayPointer != NULL)

{

numSentences++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for sentences ending with a question mark

while(arrayPointer != NULL)

{

arrayPointer = strchr(arrayPointer, '?');

if (arrayPointer != NULL)

{

numSentences++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for sentences ending with a period

while(arrayPointer != NULL)

{

arrayPointer = strchr(arrayPointer, '.');

if (arrayPointer != NULL)

{

numSentences++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for ellipses

//If there are ellipses, decrease number of sentences by 1

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, "...");

if (arrayPointer != NULL)

{

numSentences--;

arrayPointer++;

}

}

//Return the total number of sentences

return numSentences;

}

/\* Function to get the number of paragraphs

\* Takes a pointer to a char array representing the text of a speech file as

\* its parameter

\* Goes through the array and counts the number of new line characters, which

\* represent paragraphs, using a pointer variable to the array

\* Returns the total number of paragraphs as an int

\*/

int getNumParagraphs(char \*&speechString)

{

//Pointer to go through the array

char \*arrayPointer = speechString;

//Variable representing the total number of paragraphs

int numParagraphs = 0;

//Check for the occurrence of new line characters and count them

while (arrayPointer != NULL)

{

arrayPointer = strchr(arrayPointer, '\n');

if( arrayPointer != NULL)

{

numParagraphs++;

arrayPointer++;

}

}

//Return the total number of paragraphs

return numParagraphs;

}

/\* Function to get the average sentence length

\* Takes a pointer to a char array representing a speech file as its parameter

\* Calculates the average sentence length of the text by getting the number of

\* words and the number of sentences, and then dividing the number of words

\* by the number of sentences

\* If the number of sentences is 0, function returns 0

\* Returns the average sentence length as a double

\*/

double getAvgSentenceLength(char \*&speechString)

{

//Variable representing the average sentence length

double avgSentenceLength= 0;

//Get the number of words by calling the getNumWords function

int numWords = getNumWords(speechString);

//Get the number of sentences by calling the getNumSentences function

int numSentences = getNumSentences(speechString);

//Check if numSentences is 0 to avoid division by 0

if(numSentences == 0)

return avgSentenceLength;

//If numSentences isn't 0, average sentence length is the number of words

//Divided by the number of sentences

avgSentenceLength = (double)numWords/(double)numSentences;

//Return the average sentence length

return avgSentenceLength;

}

/\* Function to get the average word length

\* Takes a pointer to a char array representing a speech file as its

\* parameter

\* Calculates the average word length by getting the number of alphabetic

\* characters and the number of words, then divides the number of

\* alphabetic characters by the number of words

\* Returns the average word length as a double

\*/

double getAvgWordLength(char \*&speechString)

{

//Get the number of alphabetic characters

int numAlphabetCharacters = getNumAlphabetic(speechString);

//Get the number of words

int numWords = getNumWords(speechString);

//Average word length is the number of alphabetic characters divided by

//the number of words

double avgWordLength = (double)numAlphabetCharacters/numWords;

//Return average word length

return avgWordLength;

}

/\* Method to get the number of first-person singular pronouns

\* Takes a pointer to the speech array as its parameter

\* Counts the total number of first-person singular pronouns

\* ("I," "I'll," "I've", "me," "my" "myself," "mine")

\* by using a pointer variable to the speech array.

\* If one of the words is found, the number of sentences is increased and the

\* pointer looks at the next word in the speech array

\* Returns the total number of first-person singular pronouns

\*/

int getFirstPersonSingular(char \*&speechArray)

{

//Pointer variable used to advance through the speech array

char \*arrayPointer = speechArray;

//Variable for the total number of first-person singular pronouns

int numSingular = 0;

//Check for the occurrence of "I" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "I" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I.");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "I" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I?");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "I" at the end of a sentence with a "!"

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I!");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "I" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I,");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "I've"

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I've ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "I'll"

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " I'll ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "my" in the middle of a sentence by itself

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " my ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "my" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " My ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "mine" in the middle of a sentence by itself

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " mine ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "mine" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " mine,");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "mine" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " mine.");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "mine" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " mine?");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "mine" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " mine!");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "mine" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " Mine ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "me" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " me ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "me" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " me.");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "me" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " me!");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "me" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " me?");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Check for the occurrence of "me" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " me,");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "myself" by itself in a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " myself ");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "myself" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " myself.");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "myself" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " myself!");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "myself" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " myself?");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "myself" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " myself,");

if( arrayPointer != NULL)

{

numSingular++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Return the total number of singular pronouns

return numSingular++;

}

/\* Method to get the number of first-person plural pronouns

\* Takes a pointer to the speech array as its parameter

\* Counts the total number of first-person singular pronouns

\* ("we," "we'll," "we've", "us," "our" "ours," "ourselves")

\* by using a pointer variable to the speech array.

\* If one of the words is found, the number of sentences is increased and the

\* pointer looks at the next word in the speech array

\* Returns the total number of first-person plural pronouns

\*/

int getFirstPersonPlural(char \*&speechArray)

{

//Pointer variable used to advance through the speech array

char \*arrayPointer = speechArray;

//Variable for the total number of first-person singular pronouns

int numPlural = 0;

//Check for the occurrence of "we" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " We ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we.");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we!");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we?");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we,");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we've" in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we've ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we've" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " We've ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we'll" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " we'll ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "we'll" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " We'll ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "us" in the middle of a sentence by itself

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " us ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "us" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " Us ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "us" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " us.");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "us" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " us!");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "us" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " us?");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "us" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " us,");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "our" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " our ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "our" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " Our ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ours" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ours ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ours" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ours.");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ours" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ours!");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ours" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ours?");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ours" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " Ours ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ours" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ours,");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ourselves" by itself in the middle of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ourselves ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ourselves" at the end of a sentence with a '.'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ourselves.");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ourselves" at the end of a sentence with a '?'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ourselves?");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ourselves" at the end of a sentence with a '!'

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ourselves!");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ourselves" at the beginning of a sentence

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " Ourselves ");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Reset array pointer to go back to the beginning of the speech array

arrayPointer = speechArray;

//Check for the occurrence of "ourselves" next to a comma

while(arrayPointer != NULL)

{

arrayPointer = strstr(arrayPointer, " ourselves,");

if( arrayPointer != NULL)

{

numPlural++;

arrayPointer++;

}

}

//Return the total number of first-person plural pronouns

return numPlural;

}

/\* Function to get the ratio of first-person singular pronouns to first-person

\* plural pronouns

\* Takes a pointer to the speech array as its parameter

\* Calls the getFirstPersonSingular() and getFirstPersonPlural() methods

\* to get the total number of first-person singular pronouns and first-person

\* plural pronouns in the speech, respectively

\* Gets the ratio by dividing the number of singular pronouns by the number of

\* plural pronouns

\* If the number of plural pronouns is 0, the ratio is set to 0 to avoid

\* division by 0

\* Returns the ratio of singular to plural pronouns as a double

\*/

double getPronounRatio(char \*&speechString)

{

//Variables for the number of singular and plural pronouns

//Obtained by calling their respective functions

int numSingular = getFirstPersonSingular(speechString);

int numPlural = getFirstPersonPlural(speechString);

//Variable for the ratio of singular-plural pronouns

double ratio;

//Check if the number of plural pronouns is 0 to avoid dividing by 0

if (numPlural == 0)

ratio = 0;

//If the number of plural pronouns is not 0, the ratio is simply the number

//of singular pronouns divided by the number of plural pronouns

else

ratio = (double)numSingular/(double)numPlural;

//Return the ratio of singular pronouns to plural pronouns

return ratio;

}

/\* Method to set the file names for each speech

\* Takes an array of 57 speech structs as its parameter

\* Hard codes the speechFileName component of an array of speech structs

\* to the associated file name in the directory

\* Each file name is in the format "directory/year\_presidentName.txt"

\*/

void setFileNames(Speech speeches[57])

{

strcpy(speeches[0].speechFileName, "datafiles\1789\_George\_Washington.txt");

strcpy(speeches[1].speechFileName, "datafiles\1793\_George\_Washington.txt");

strcpy(speeches[2].speechFileName, "datafiles\1797\_John\_Adams.txt");

strcpy(speeches[3].speechFileName, "datafiles\1801\_Thomas\_Jefferson.txt");

strcpy(speeches[4].speechFileName, "datafiles\1805\_Thomas\_Jefferson.txt");

strcpy(speeches[5].speechFileName, "datafiles\1809\_James\_Madison.txt");

strcpy(speeches[6].speechFileName, "datafiles\1813\_James\_Madison.txt");

strcpy(speeches[7].speechFileName, "datafiles\1817\_James\_Monroe.txt");

strcpy(speeches[8].speechFileName, "datafiles\1821\_James\_Monroe.txt");

strcpy(speeches[9].speechFileName, "datafiles\1825\_John\_Quincy\_Adams.txt");

strcpy(speeches[10].speechFileName, "datafiles\1829\_Andrew\_Jackson.txt");

strcpy(speeches[11].speechFileName, "datafiles\1833\_Andrew\_Jackson.txt");

strcpy(speeches[12].speechFileName, "datafiles\1837\_Martin\_Van\_Buren.txt");

strcpy(speeches[13].speechFileName, "datafiles\1841\_William\_Henry\_Harrison.txt");

strcpy(speeches[14].speechFileName, "datafiles\1845\_James\_Polk.txt");

strcpy(speeches[15].speechFileName, "datafiles\1849\_Zachary\_Taylor.txt");

strcpy(speeches[16].speechFileName, "datafiles\1853\_Franklin\_Pierce.txt");

strcpy(speeches[17].speechFileName, "datafiles\1857\_James\_Buchanan.txt");

strcpy(speeches[18].speechFileName, "datafiles\1861\_Abraham\_Lincoln.txt");

strcpy(speeches[19].speechFileName, "datafiles\1865\_Abraham\_Lincoln.txt");

strcpy(speeches[20].speechFileName, "datafiles\1869\_Ulysses\_Grant.txt");

strcpy(speeches[21].speechFileName, "datafiles\1873\_Ulysses\_Grant.txt");

strcpy(speeches[22].speechFileName, "datafiles\1877\_Rutherford\_Hayes.txt");

strcpy(speeches[23].speechFileName, "datafiles\1881\_James\_Garfield.txt");

strcpy(speeches[24].speechFileName, "datafiles\1885\_Grover\_Cleveland.txt");

strcpy(speeches[25].speechFileName, "datafiles\1889\_Benjamin\_Harrison.txt");

strcpy(speeches[26].speechFileName, "datafiles\1893\_Grover\_Cleveland.txt");

strcpy(speeches[27].speechFileName, "datafiles\1897\_William\_McKinley.txt");

strcpy(speeches[28].speechFileName, "datafiles\1901\_William\_McKinley.txt");

strcpy(speeches[29].speechFileName, "datafiles\1905\_Theodore\_Roosevelt.txt");

strcpy(speeches[30].speechFileName, "datafiles\1909\_William\_Taft.txt");

strcpy(speeches[31].speechFileName, "datafiles\1913\_Woodrow\_Wilson.txt");

strcpy(speeches[32].speechFileName, "datafiles\1917\_Woodrow\_Wilson.txt");

strcpy(speeches[33].speechFileName, "datafiles\1921\_Warren\_Harding.txt");

strcpy(speeches[34].speechFileName, "datafiles\1925\_Calvin\_Coolidge.txt");

strcpy(speeches[35].speechFileName, "datafiles\1929\_Herbert\_Hoover.txt");

strcpy(speeches[36].speechFileName, "datafiles\1933\_Franklin\_Roosevelt.txt");

strcpy(speeches[37].speechFileName, "datafiles\1937\_Franklin\_Roosevelt.txt");

strcpy(speeches[38].speechFileName, "datafiles\1941\_Franklin\_Roosevelt.txt");

strcpy(speeches[39].speechFileName, "datafiles\1945\_Franklin\_Roosevelt.txt");

strcpy(speeches[40].speechFileName, "datafiles\1949\_Harry\_Truman.txt");

strcpy(speeches[41].speechFileName, "datafiles\1953\_Dwight\_Eisenhower.txt");

strcpy(speeches[42].speechFileName, "datafiles\1957\_Dwight\_Eisenhower.txt");

strcpy(speeches[43].speechFileName, "datafiles\1961\_John\_Kennedy.txt");

strcpy(speeches[44].speechFileName, "datafiles\1965\_Lyndon\_Johnson.txt");

strcpy(speeches[45].speechFileName, "datafiles\1969\_Richard\_Nixon.txt");

strcpy(speeches[46].speechFileName, "datafiles\1973\_Richard\_Nixon.txt");

strcpy(speeches[47].speechFileName, "datafiles\1977\_Jimmy\_Carter.txt");

strcpy(speeches[48].speechFileName, "datafiles\1981\_Ronald\_Reagan.txt");

strcpy(speeches[49].speechFileName, "datafiles\1985\_Ronald\_Reagan.txt");

strcpy(speeches[50].speechFileName, "datafiles\1989\_George\_Bush.txt");

strcpy(speeches[51].speechFileName, "datafiles\1993\_Bill\_Clinton.txt");

strcpy(speeches[52].speechFileName, "datafiles\1997\_Bill\_Clinton.txt");

strcpy(speeches[53].speechFileName, "datafiles\2001\_George\_W\_Bush.txt");

strcpy(speeches[54].speechFileName, "datafiles\2005\_George\_W\_Bush.txt");

strcpy(speeches[55].speechFileName, "datafiles\2009\_Barack\_Obama.txt");

strcpy(speeches[56].speechFileName, "datafiles\2013\_Barack\_Obama.txt");

}

/\* Method to set the presidet name for each speech

\* Takes an array of 57 speech structs as its parameter

\* Hard Codes the first and last name of each president that gave each speech

\* and sets it tothe .presidentName element of each struct

\*/

void setPresidentNames(Speech speeches[57])

{

strcpy(speeches[0].presidentName, "George Washington");

strcpy(speeches[1].presidentName, "George Washington");

strcpy(speeches[2].presidentName, "John Adams");

strcpy(speeches[3].presidentName, "Thomas Jefferson");

strcpy(speeches[4].presidentName, "Thomas Jefferson");

strcpy(speeches[5].presidentName, "James Madison");

strcpy(speeches[6].presidentName, "James Madison");

strcpy(speeches[7].presidentName, "James Monroe");

strcpy(speeches[8].presidentName, "James Monroe");

strcpy(speeches[9].presidentName, "John Quincy Adams");

strcpy(speeches[10].presidentName, "Andrew Jackson");

strcpy(speeches[11].presidentName, "Andrew Jackson");

strcpy(speeches[12].presidentName, "Martin Van Buren");

strcpy(speeches[13].presidentName, "William Henry Harrison");

strcpy(speeches[14].presidentName, "James Polk");

strcpy(speeches[15].presidentName, "Zachary Taylor");

strcpy(speeches[16].presidentName, "Franklin Pierce");

strcpy(speeches[17].presidentName, "James Buchanan");

strcpy(speeches[18].presidentName, "Abraham Lincoln");

strcpy(speeches[19].presidentName, "Abraham Lincoln");

strcpy(speeches[20].presidentName, "Ulysses S. Grant");

strcpy(speeches[21].presidentName, "Ulysses S. Grant");

strcpy(speeches[22].presidentName, "Rutherford B. Hayes");

strcpy(speeches[23].presidentName, "James Garfield");

strcpy(speeches[24].presidentName, "Grover Cleveland");

strcpy(speeches[25].presidentName, "Benjamin Harrison");

strcpy(speeches[26].presidentName, "Grover Cleveland");

strcpy(speeches[27].presidentName, "William McKinley");

strcpy(speeches[28].presidentName, "William McKinley");

strcpy(speeches[29].presidentName, "Theodore Roosevelt");

strcpy(speeches[30].presidentName, "William Taft");

strcpy(speeches[31].presidentName, "Woodrow Wilson");

strcpy(speeches[32].presidentName, "Woodrow Wilson");

strcpy(speeches[33].presidentName, "Warren Harding");

strcpy(speeches[34].presidentName, "Calvin Coolidge");

strcpy(speeches[35].presidentName, "Herbert Hoover");

strcpy(speeches[36].presidentName, "Franklin D. Roosevelt");

strcpy(speeches[37].presidentName, "Franklin D. Roosevelt");

strcpy(speeches[38].presidentName, "Franklin D. Roosevelt");

strcpy(speeches[39].presidentName, "Franklin D. Roosevelt");

strcpy(speeches[40].presidentName, "Harry S. Truman");

strcpy(speeches[41].presidentName, "Dwight D. Eisenhower");

strcpy(speeches[42].presidentName, "Dwight D. Eisenhower");

strcpy(speeches[43].presidentName, "John F. Kennedy");

strcpy(speeches[44].presidentName, "Lyndon B. Johnson");

strcpy(speeches[45].presidentName, "Richard Nixon");

strcpy(speeches[46].presidentName, "Richard Nixon");

strcpy(speeches[47].presidentName, "Jimmy Carter");

strcpy(speeches[48].presidentName, "Ronald Reagan");

strcpy(speeches[49].presidentName, "Ronald Reagan");

strcpy(speeches[50].presidentName, "George Bush");

strcpy(speeches[51].presidentName, "Bill Clinton");

strcpy(speeches[52].presidentName, "Bill Clinton");

strcpy(speeches[53].presidentName, "George W. Bush");

strcpy(speeches[54].presidentName, "George W. Bush");

strcpy(speeches[55].presidentName, "Barack Obama");

strcpy(speeches[56].presidentName, "Barack Obama");

}

/\* Method to set the year for each speech

\* Takes an array of 57 speech structs as its parameter

\* Hard Codes the year each speech was given to the .year element of each

\* struct

\*/

void setYears(Speech speeches[57])

{

speeches[0].year = 1789;

speeches[1].year = 1793;

speeches[2].year = 1797;

speeches[3].year = 1801;

speeches[4].year = 1805;

speeches[5].year = 1809;

speeches[6].year = 1813;

speeches[7].year = 1817;

speeches[8].year = 1821;

speeches[9].year = 1825;

speeches[10].year = 1829;

speeches[11].year = 1833;

speeches[12].year = 1837;

speeches[13].year = 1841;

speeches[14].year = 1845;

speeches[15].year = 1849;

speeches[16].year = 1853;

speeches[17].year = 1857;

speeches[18].year = 1861;

speeches[19].year = 1865;

speeches[20].year = 1869;

speeches[21].year = 1873;

speeches[22].year = 1877;

speeches[23].year = 1881;

speeches[24].year = 1885;

speeches[25].year = 1889;

speeches[26].year = 1893;

speeches[27].year = 1897;

speeches[28].year = 1901;

speeches[29].year = 1905;

speeches[30].year = 1909;

speeches[31].year = 1913;

speeches[32].year = 1917;

speeches[33].year = 1921;

speeches[34].year = 1925;

speeches[35].year = 1929;

speeches[36].year = 1933;

speeches[37].year = 1937;

speeches[38].year = 1941;

speeches[39].year = 1945;

speeches[40].year = 1949;

speeches[41].year = 1953;

speeches[42].year = 1957;

speeches[43].year = 1961;

speeches[44].year = 1965;

speeches[45].year = 1969;

speeches[46].year = 1973;

speeches[47].year = 1977;

speeches[48].year = 1981;

speeches[49].year = 1985;

speeches[50].year = 1989;

speeches[51].year = 1993;

speeches[52].year = 1997;

speeches[53].year = 2001;

speeches[54].year = 2005;

speeches[55].year = 2009;

speeches[56].year = 2013;

}

/\* Function to read the speech text from a file into an array

\* Takes an array of speech structs and an integer representing an element of

\* that array as its parameters

\* Opens each speech file based on the struct's .speechFileName element

\* After opening the file, reads all of the characters in the file and sets

\* them to the .speechArray element of the struct, representing the actual text

\* of the speech.

\*/

void readSpeechFile(Speech speeches[], int speechNum)

{

//Pointer to the speech file

FILE\* pInputFile;

//Associate the actual file name with file pointer and try to open it

pInputFile = fopen(speeches[speechNum].speechFileName, "r");

//Check if the file is null

if(pInputFile == NULL)

{

printf("Attempt to open file %s failed. Exiting ... \n", pInputFile);

exit(-1);

}

//Char variable to move to the next character in each file

char nextChar = getc(pInputFile);

//Variable to represent the number of characters in each file

int numCharacters = 0;

//Count the number of characters in each file

while (nextChar != EOF)

{

numCharacters++;

nextChar = getc(pInputFile);

}

//Loop counter

int j = 0;

//Reopen the file to set the text from the file to the speech struct

freopen(speeches[speechNum].speechFileName,"r", pInputFile);

//Dynamically allocate memory for the speechArray element of the struct

//Based on the number of characters in the file

speeches[speechNum].speechArray = new (char[numCharacters + 1]);

//Read the characters from the file into the new array

while(fscanf(pInputFile, "%c", &speeches[speechNum].speechArray[j]) != EOF)

j++;

//Close the file

fclose(pInputFile);

}

//Main method

int main()

{

//Prints project header with name, lab time, and project name

printf("\nAllison Channic\n");

printf("Lab: Thursday 12:00pm\n\n");

printf("Lily Lu\n");

printf("Lab: Thursday 2:00pm\n\n");

printf("Program #4: Analyze This\n\n\n");

//Initialize the speech files into an array of speech structs

Speech speeches[57];

//Set file names

setFileNames(speeches);

//Set the president names

setPresidentNames(speeches);

//Set years each speech was given

setYears(speeches);

//Initialize variables for data

int numWords = 0;

int numSentences = 0;

int numParagraphs = 0;

double avgWordLength = 0;

double avgSentenceLength = 0;

int numSingular = 0;

int numPlural = 0;

double pronounRatio = 0;

//Print column headers for the table

printf(" President");

printf("\t\t Year");

printf(" WC");

printf(" SC");

printf(" PC");

printf(" AWL");

printf(" ASL");

printf(" FPSP");

printf(" FPPP");

printf(" SP/PP");

printf("\n");

//Print line to separate column headers and data

int i = 0;

while (i < 80)

{

printf("\_");

i++;

}

//Get the data for each of the 57 speeches

for (i = 0; i < 57; i++)

{

//Read the text from the speech file into the array

readSpeechFile(speeches, i);

//Set the variables for data

//Get the number of words

numWords = getNumWords(speeches[i].speechArray);

//Get the number of sentences

numSentences = getNumSentences(speeches[i].speechArray);

//Get the number of paragraphs

numParagraphs = getNumParagraphs(speeches[i].speechArray);

//Get the average word length

avgWordLength = getAvgWordLength(speeches[i].speechArray);

//Get the average sentence length

avgSentenceLength = getAvgSentenceLength(speeches[i].speechArray);

//Get the number of first person singular pronouns

numSingular = getFirstPersonSingular(speeches[i].speechArray);

//Get the number of first person plural pronouns

numPlural = getFirstPersonPlural(speeches[i].speechArray);

//Get the ratio of first person singular pronouns to first person

//plural pronouns

pronounRatio = getPronounRatio(speeches[i].speechArray);

//Print the number of each speech

if ((i+1) < 10)

printf(" %d.) ", i+1);

else

printf("%d.) ", i+1);

//Print out the data in a table format

printf("%-23s", speeches[i].presidentName);

printf("%d", speeches[i].year);

printf("%6d", numWords);

printf("%5d", numSentences);

printf("%5d", numParagraphs);

printf("%6.2lf", avgWordLength);

printf("%6.1lf", avgSentenceLength);

printf("%5d", numSingular);

printf("%5d", numPlural);

if (numPlural == 0)

printf(" N/A");

else

printf("%9.2lf", pronounRatio);

printf("\n");

}

}