Report

1. There were many obstacles over the course of completing this project. For one, I had to figure out how to remove two incorrect elements in a row in makeProper. In order to do this, I ended up having to decrement the for loop counter in order to check the same position again (so as to not move past an incorrect position). Another major problem I had was holding onto the values within a 2d array from the previous call. In order to fix this, I cleared the 2d array of word documents by setting every element to the zero byte. Another problem was figuring out how to calculate the rate accounting for the matching word being before or after the first word of the pattern. In order to fix this, I changed my code to allow for it to check both word1 and word2 arrays and only check the words after. Finally, I was having major problems with accessing elements outside the bounds of the array. There was not quick fix to this, I just had to keep printing out values until I understood what was wrong with the code.

Pseudocode

**Eliminate- a helper function that would “remove” a row from the 3 arrays passed in**

Take an index to start removing values from

Start moving all values from one row further to the earlier row until one less the number of patterns

**MakeProper- returns the number of valid patterns in the passed in arrays**

Have a for loop that converts all alphabetic characters in word1 and word2 to lower case

Have a for loop that removes all rows with negative separation values

If a negative separation value is found, call eliminate, decrease nPatters and then decrement the index by 1 so it doesn’t skip checking any separation values

Have a for loop that removes all rows with non-alphabetic characters in word1

If a negative separation value is found, call eliminate, decrease nPatters and then decrement the index by 1 so it doesn’t skip checking any words

Have a for loop that removes all rows with non-alphabetic characters in word2

If a negative separation value is found, call eliminate, decrease nPatters and then decrement the index by 1 so it doesn’t skip checking any words

Have a for loop that removes all empty strings in word1

If a negative separation value is found, call eliminate, decrease nPatters and then decrement the index by 1 so it doesn’t skip checking any words

Have a for loop that removes all empty strings in word2

If a negative separation value is found, call eliminate, decrease nPatters and then decrement the index by 1 so it doesn’t skip checking any words

Have a for loop that starts at 0 and goes up to the value of nPatterns

Have a nested for loop that starts at the previous index plus one

If the pairs are found in the opposite order in word1 and word2 or if they’re found in the same order in word1 and word2

Check which index has a great separation value

Remove accordingly, decrease nPatterns, and decrement the correct index based on which row was removed

**Rate- determines the number of patterns found within a document**

Load the document array into another array (fixedDocument) that isn’t constant

Determine what index the document stops at, and make the next value in the fixedDocument a zero byte

Convert all characters in the fixedDocument array into lowercase

Have a for loop that removes all characters not alphabets or spaces

Decrements index of for loop to not skip and characters

Have a for loop that removes consecutive spaces

Checks if the index and the index + 1 element of fixedDocument is a space

If so, copy the array over one of the spaces

Decrease index by 1

Have a for loop that counts the number of spaces

Initialize a 2D array called words of document- make it size 251 x 251 just to be safe

Have a for loop that loads the fixedDocument contents into the 2D array

If it’s not a space, initialize the character into the given row and column, and then increment the column index

If it is a space, set the given row’s next element to the zero byte, increment the row, and then reset the column index to be 0

Have a for loop going through the number of patterns

Have a nested for loop that goes through the number of rows

If a word in wordsOfDocument are also in word1

Check if the word2 is found within the separation value of that nPattern row

If the word is found, increment counter, and break out of while loop and nested for loop

If no word is found up to separation values, exit the while loop

Make sure there is no index out of bounds

If a word in wordsOfDocument are also in word2

Check if the word1 is found within the seperation value of that nPattern row

If the word is found, increment counter, and break out of while loop and nested for loop

If no word is found up to separation values, exit the while loop

Make sure there is no index out of bounds

Have a for loop that clears the entire array by setting every character to a zero byte (nested for loop needed)