**Lab Report 01**

Problem

In this lab, we are tasked with creating a sorting algorithm for an array of words based on each word’s number of vowels, consonants, and length

Lab Solution

Each sorting method copies the original array of words to be used in sorting. The vowel and consonant methods create an additional new array that stores the number of vowels and consonants to be used for sorting purposes. The original text file array is printed on the console alongside the 3 new sorted ones.

Problems Encountered

At first, I tried to use a bubble sorting algorithm, but my code would run indefinitely for a reason unknown to me. What I do know is that it has to do with the array being full of Strings, since it works perfectly fine with integers. I also had trouble iterating through the array to retrieve all the vowel and consonant amounts at first. I eventually figured out storing the values in a separate array worked best for comparison purposes.

4. Arrays work perfectly for cases where you need to iterate through a list of items for sorting or storing values.

5. It can be tedious to set up properly and isn’t necessarily always the best to use in some cases

6. the index 0 refers to the 1st item in an array, which mean that valid indices are between 0 and the literal last index minus 1

7. Within the first loop, an error occurs when define the character at each index because you can’t add a char and int value together. Instead, you could copy an array of char values that iterate from a-z.

8. The for loop will not iterate properly because the third statement is incorrect; changing it to i++ will fix this. Change the else if to an if statement to ensure that both statements execute regardless of each other.

9. The method will copy and return the memory address for the array instead of its values. To fix this, a for loop that iterates through the length of the array may be used. Each index of the new array will be equal to the value of the old one and will return as such.

10.

1.0

2.0

3.0

4.0

5.0

2.0

4.0

6.0

8.0

10.0