Word Frequency Specs

• The program should allow the user to enter a single-character base input and single-character second input, compare if they match, and return the number of matches found.

⁃ Input: ‘a’ (base input); ‘a’ (second input) — testing for successful single character match

⁃ Output: 1

• The program should allow the user to enter a two-character base input and a two-character second input, compare if they match, and return the number of matches found.

⁃ Input: ‘cd’ (base input); ‘cd’ (second input) — testing for successful multi-character match

⁃ Output: 1

• The program should allow the user to enter at least a single-character base input and a multi-character second input, compare if they fully match, and return the number of matches found.

⁃ Input: ‘cat’ (base input); ‘catty’ (second input) — testing that the program is able to hold the entire second input in its original value and distinguish it as not matching the entire value of the base input

⁃ Output: (null)

• The program should allow the user to enter a base input and a string of at least two words for second input, compare the base input to the entire string, and return the number of matches found.

⁃ Input: ‘be’ (base input); ‘bee is be’ (second input) — testing that the program can compare a single-word base input string to a multi-word second input string and find the exact word match

⁃ Output: 1

• The program should allow the user to enter a base input with upper or lowercase letters, and a multi-word string second input with upper or lowercase letters, convert all inputs to lowercase, compare the converted base input to the converted entire second input, and return the number of matches found.

⁃ Input: ‘Bird’ (base input); ‘birD is the word’ (second input) — testing for successful conversion of inputs to lowercase letters for proper comparison

⁃ Output: 1

• The program should allow the user to enter a base input and a string of at least two matching words for second input, compare the base input to the entire string, and return the number of matches found.

⁃ Input: ‘tree’ (base input); ’Tree is trees is tree’ (second input) — testing for successful match of both matching words from the string

⁃ Output: 2