1. Create 4 global variables
 1a. counter that has an int data type
 1b. wordChecker that has an int data type
 1c. corrections that has an int data type
 1d. fixedString that has a string data type
2. Have the more_to_read function that was already created
3. Create a compare_files function with a int return type
 parameters:
  - 3 string vectors named fixMeVector, typoVector, and fixoVector
  - a string variable named fixedSentence
 3a. Initialize 3 of the global variables (counter, wordChecker, fixedSentence)
 3b. Loop through the fixMeVector and typoVector
 3c. Compare the items in each vector with an if statement
  -if the items match then increment both the counter and wordChecker variables by 1 and append the fixo word to th
e fixedSentence variable
 3d. Outside of the for loops add an if statement that checks if wordChecker is not equal to 1
  -if its not equal to 1 then append the fixMeWord item to the fixedSentence variable
 3e. Reset the wordChecker variable to 0
 3f. Return the counter variable
4. Create a loop_through_files function with a void return type
 parameters:
  - 3 files (fixoFile, typoFile, and fixMeFile)
  - 3 string vectors (fixoVector, typoVector, and fixMeVector)
  - an int variable named corrections
 4a. Set up a while loop that runs while there is still more to read in both the fixoFile and typoFile (use the more_to_r
ead function)
  -within the loop initialize the fixo and typo string variables, read data from the fixo and typo file, and add that data t
o the fixo and typo vectors
 4b. Set up another while loop that runs while there is still more to read in the fixMeFile (use the more_to_read funct
ion)
  -within the loop initialize the fixMeString variable, read data from the fixMeFile, and add that data to the fixMeVec
tor
 4c. Call the compare_files variable with all of the vectors and string variable and save it into the corrections variable
5. Create a check_files function with a bool return type
 parameters:
  - 2 files (fixoFile and typoFile)
 5a. Setup up an if statement
  -returns true if there is more to read in either the fixoFile or the typoFile (use the more_to_read function), returns fa
lse if there isn't
6. Create a remove_end_space function with a string return type
 parameters:
  - a string variable (string)
 6a. Initialize a lastCharacter variable to the index of the last character in the string argument
 6b. Set up a while loop that runs while the lastCharacter variable is greater than or equal to 0 and the character at the
 given index is blank
  -subtract the lastCharacter variable by 1
 6c. Return the substring of the original string without the last character
7. Use the fcc function with the given parameters
 7a. initilaize the fix, typo, and fixMe files and open all of them
 7b. Initilaize 3 vectors (fixoWords, typoWords, and fixMeWords)
 7c. Call the loop_through_files function with the given parameter variables/files/vectors
 7d. Call the remove_end_spaces function with fixedSentence as the argument. Save it to the fixed_sentence variable
 and add a period at the end
 7e. Set up an if statement that calls the check_files function with the fix and typo files as parameters
  -if there is still more data then it will return -1, if not then it will return the number of corrections