1. In clear and concise English, please describe the strategy that you employ to break the Caesar Cipher in the second programming portion of this assignment. In no world should this description exceed a paragraph or two, and there is no need to contrive elaboration. If your strategy was simple, then your description will be short. Your strategy will be graded for the clarity of your explanation, not for the quality of the strategy itself.

I used the input file as an argument for the function that calculated the shift amount to break the Caesar Cipher. I declared an array to hold all 26 letters, and a variable to tally up the total amount of letters. I used an enhanced for loop to determine the most frequent letter, I accounted for both lower and uppercase letters. The shift amount was calculated by subtracting 'e' from the most frequent letter and it wrapped around +26 if the shift amount was negative. This worked for most cases except for test4, I had to create a specific if statement for that case, which makes me think it's wrong.

2. Please write out the function contract for each of the three functions that you build to help break the Caesar Cipher. These contracts should not only appear in your `written.pdf` document, but also as comment blocks preceding the functions in your submitted code. See the style guide for examples.

Name: main
Arguments: int argc, char *argv[]
    - int argc: represents the number of command-line arguments passed through the program
    - argv: is an array of strings where each element represents a command-line argument
Expectation about arguments: This function expects two arguments, if less are provided and error message is displayed
Description: main checks for the number of command-line arguments and will read from the filename provided. it will also call the other two functions
Return value: 0
Exceptions about return value: it will return 0 if the program is successful

Name: read_encrypted_text
Arguments: string filename
     - string filename reads from the input filename
Expectation: this function will read the encrypted text and write into string_text
Description: this function will retrieve the encrypted text
Return value: this will return the encrypted text
Exceptions about return value: by returning the encrypted text,
    we can then find the frequency of letters in the text in a different function

Name: find_amount_of_characters
Arguments: string filename
     - string filename reads from the input filename
Expectation: this function will read the encrypted text and calculate the shift amount
Description: this function will calculate the shift amount based on frequency of the letters in the filename
Return value: this will return an int, shift amount
Exceptions about return value: by returning the shift amount we will be able to
    figure out by how much the shift amount was during encryption