CSC 401 ASSIGNMENT FOUR

Due Date: Tuesday, Oct. 3rd by 11:58 PM

The purpose of this assignment is to assess your understanding of

- Creating user-defined functions, passing and returning one or more variables
- Utilizing FOR Loop iteration patterns
- Reading and Writing text files

SUBMISSION

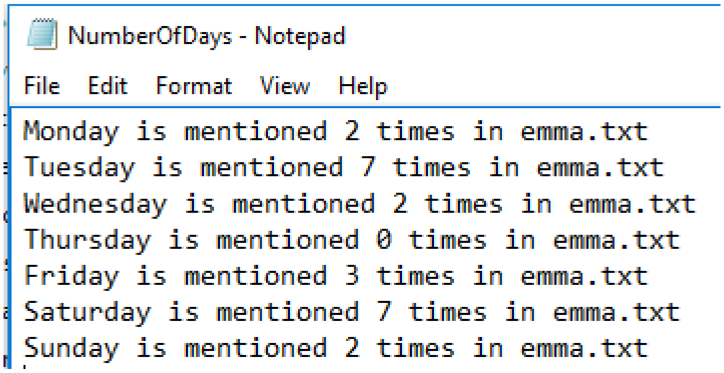
- **Include your full name as a comment in the first line of your Python program**
- **Include the problem number as a comment in the second line of our Python program**
- Save each program to a separate file labeled as YourName_hw4_1.py, YourName_hw4_2.py
- Upload each file to Submissions folder in D2L.

PROBLEMS

Note: you may not use Python statements, functions, data types, etc. that were not discussed in the reading assignment or the lecture notes/videos for this week or previous weeks. This is a class for students who have not programmed before and I expect everyone to code on the same level. If you have a better way of writing the code, then upload two versions: one that codes according to the specifications and the other that demonstrates advanced programming techniques.

I encourage you to use computational thinking to solve the problems. These are straight-forward solutions, but developing a good habit of analyzing the problem and describing the steps will serve you well as the problems get more complex.

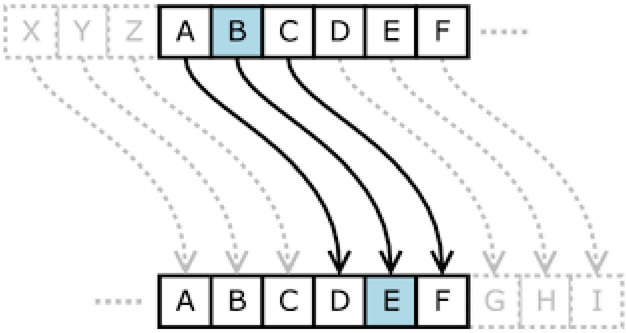
1. (File Input/Output, FOR, function 10 points) Write a function getCount(fname) that counts how often each weekday is mentioned in a text file with name fname, and writes the results to a text file with name NumberOfDays.txt. For example, I ran getCount('emma.txt') and it created the following text-file.



```
Monday is mentioned 2 times in emma.txt
Tuesday is mentioned 7 times in emma.txt
Wednesday is mentioned 2 times in emma.txt
Thursday is mentioned 0 times in emma.txt
Friday is mentioned 3 times in emma.txt
Saturday is mentioned 7 times in emma.txt
Sunday is mentioned 2 times in emma.txt
```

You are to use Pride_and_Prejudice.txt (the same file that you used in assignment three). Do not be concerned with punctuation. Use the days of the week as written in the sample output. Do NOT create seven separate write statements.

2. (File Input/Output, FOR, function 40 points) In cryptography, a Caesar cipher is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the original text is replaced by a letter some fixed number of positions down the alphabet. For example, with a right shift of 3, A would be replaced with D, B would be replaced with E, and so on



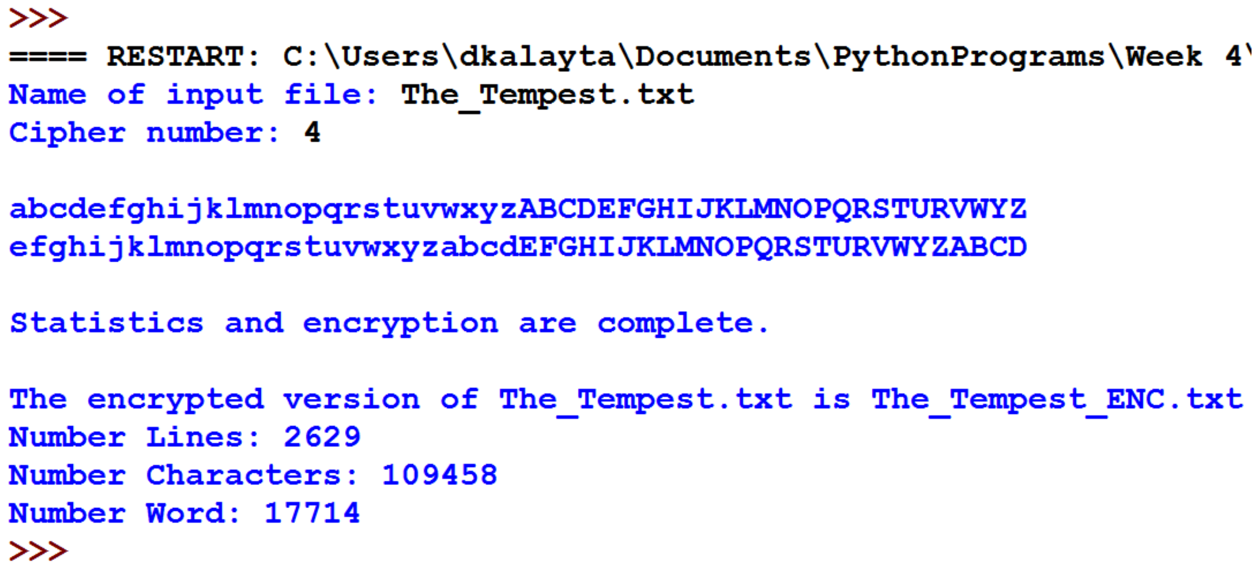
(Wikipedia, https://en.wikipedia.org/wiki/Caesar_cipher). The method is named after Julius Caesar who used it in his private correspondence.

- Write a function processFile(fname, num) that:
  - takes two parameters.
    - fname represents the name of the file to be encrypted.
    - num is an integer parameter that indicates the desired shift in the alphabet to be used in the cipher (3 in the above example). You must allow for this integer to be any value. If an integer greater than 25 is entered, you must go back to the beginning of the alphabet and keep counting. For example, if num is 26, then no shift will be performed; it is as if 0 were entered. If num is 28, then 'a' will be replaced with 'c'; as if 2 were entered. (use remainder operator; num%26 to determine the actual shift value.)
  - Count the number of lines, the number of words, and the number of characters in the input file. For our purposes, a 'word' is anything between two spaces. Do not be concerned with punctuation marks (use FOR loop with count/accumulator)
  - Generate a Caesar encrypted output file using the provided cypher (shift) value. The name of the output file should be 'inputFileName_ENC.txt' (use string slicing and concatenation to create the new output file name; did not hard code the output file name.)
  - Lower and Upper case letters must be respected, in the encryption process. Lower case letters map to the cipher's lower case letters; upper case letters map to the cipher's upper case letters. Hint: create 2 strings; one with the

lower case letters of the alphabet and the other with the upper case letters of the alphabet.

- Create a string for the cipher; slice the alpha strings you created above based on the desired shift; concatenate all sliced strings to create the cipher.
- You must open the input file only once in your function.
- Do not forget to close your files.
- Do all processing on a line of a file
- A function can return more than one result at a time. If you need to return x, y and z, use return (x,y,z). Your function must return the name of the input file, output file, number of lines, number of words, and number of characters in the input file.

- In the same file where you wrote the processFile() function, write the code that does the following:
  - Ask the user for the name of the file to be encrypted.
  - Ask the user for the cipher (that is, the number of characters to be shifted)
  - Call the processFile() function with the data entered by the user
  - Print the input and output file names, the number of lines, the number of characters and the number of words for the input file.
- Sample execution:



```
>>>
==== RESTART: C:\Users\dkalayta\Documents\PythonPrograms\Week 4\
Name of input file: The_Tempest.txt
Cipher number: 4

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTURVWYZ
efghijklmnopqrstuvwxyzabcdEFGHIJKLMNOPQRSTURVWYZABCD

Statistics and encryption are complete.

The encrypted version of The_Tempest.txt is The_Tempest_ENC.txt
Number Lines: 2629
Number Characters: 109458
Number Word: 17714
>>>
```

IF YOU HAVE ANY QUESTIONS REGARDING THIS ASSIGNMENT, PLEASE POST THEM TO THE ASSIGNMENT FOUR DISCUSSION FORUM.