CMPSC-132: Programming and Computation II
Spring 2019

Lab #4

Due Date: 02/01/2019, 11:59PM

Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.

Instructions:

- The work in this lab must be completed alone and must be your own.
- **Download the starter code file from the LAB4 Assignment on Canvas. Do not change the function names or given started code on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class.
- Each function must return the output (Do not use print in your final submission, otherwise your submissions will receive a -1 point deduction)
- Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.

Goal:

[6 pts] Write the functions *encrypt*(message, key) and *decrypt*(message, key), where *message* is a string and *key* is a positive integer. Both functions **return** a string that contains *message* encrypted/decrypted using the Caesar cipher method. If a string is not provided as an input for message or an integer is not provided for key, the function must **return** a string with an error message

- Functions must encrypt/decrypt both uppercase and lowercase letters
- decrypt(message, key) retrieves the original message encrypted by the encrypt function if the same key is used
- Numbers and punctuation can remain the same

[4 pts] Write the unittest script to perform your testing for both functions *encrypt* and *decrypt* using the unittest module and prove that they work properly. Remember to be as descriptive as possible and write as many cases as necessary (check the Hands On video for example of a description). Unittest is intended to run test cases in bulk, so it should contain enough test cases to prove your code works. Discuss possible edge cases during your recitation. Do not include the doctest cases

Deliverables:

- Submit your encrypt and decrypt code in a file named LAB4.py to the Lab4 **GradeScope** assignment before the due date
- Submit your unittest code in a file named test.py to the Lab4 **CANVAS** assignment before the due date

Notes:
One of the purposes of unit testing is to provide evidence that everything was tested and it works properly. Feedback will be provided if your code does not work according to the assignment requirements, but no partial credit will be given. You can use https://www.braingle.com/brainteasers/codes/caesar.php#form to create test cases