

## HW9 Cipher in Python

Due: December 3, 2017

Your task is to create a Python program called `my_cipher.py`. In this assignment, you will need to implement a program that performs an encryption on the contents of a file `message.txt` with a transformed alphabet.

Given a keyword input by the user, a transformed alphabet can be created in the following process. Traverse through the keyword's letters. If a letter has appeared in previous part of the keyword (i.e., a repeated letter), remove it. Now you get a simplified keyword. Second, remove the letters included in the simplified keyword from the alphabet, deriving a simplified alphabet. Third, concatenate the simplified keyword and the simplified alphabet. The result is actually a transformed alphabet. For example, if the keyword is "computer", then you have the following:

Alphabet:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Transformed Alphabet:	COMPUTERABDFGHIJKLNQSVWXYZ

For another example, if the keyword is "hello", then you have the following:

Alphabet:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Transformed Alphabet:	HELOABCDEFGHIJKMNPQRSTUVWXYZ

With the transformed alphabet, the encryption algorithm simply substitutes each letter in the to-be-encrypted message with its counterpart (which has the same position/index) in the transformed alphabet. If there is a space, the space is kept in the ciphertext in the same position/index. The encryption result over a message (also called plaintext) is called ciphertext. For example, under the second transformed alphabet above, the ciphertext of plaintext "HI RE" should be "DF RA", where H (the 8<sup>th</sup> letter in Alphabet) is substituted by D (the 8<sup>th</sup> letter in Transformed Alphabet), I (the 9<sup>th</sup> letter in Alphabet) is substituted by F (the 9<sup>th</sup> letter in Transformed Alphabet), space is just space in the ciphertext, R (the 18<sup>th</sup> letter in Alphabet) is substituted by R (the 18<sup>th</sup> letter in Transformed Alphabet), and E (the 5<sup>th</sup> letter in Alphabet) is substituted by A (the 5<sup>th</sup> letter in Transformed Alphabet).

Your program needs to prompt the user for a keyword. After the user inputs the keyword, your program should output the user's chosen keyword, the original alphabet, and the transformed alphabet to the screen. They should all in upper-case letters no matter if the user input includes lower-case letters or not. For the contents of `message.txt`, take each line as a plaintext, encrypt the line, and append the ciphertext as a line into `message.txt`. All the appended ciphertext lines must be upper-case letters. The process of deriving a transformed alphabet from the original alphabet and the keyword should be implemented as a function.

Here's an example execution. Let's say the `message.txt` has the following contents "Hi RA" in one line, and the user input keyword is "hello". Your program should produce the following output to the screen:

Keyword: HELLO

Alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Transformed Alphabet: HELOABCDFGIJKMNPQRSTUVWXYZ

Your program should also append one line to the message.txt, making it's content to be:

Hi RE

DF RA

For submission, just submit the .py file. The graders will create the message.txt file when grading.

**Grading rubric:**

- Running error caused by code: -90%
- Not prompting user for keyword but working for any hardcoded keyword: -20%
- Not printing keyword, alphabet, and transformed alphabet to screen: -20%
- Incorrect transformed alphabet: -70%
- Lower-case and upper-case issues: -30%
- Correct transformed alphabet but incorrect ciphertext: -40%
- Hardcoded transform and encryption results: -90%
- Not working with message.txt but a different file name: -20%
- Only able to handle one line in message.txt but not multiple lines: -20%
- Alphabet transformation is not implemented as a function: -20%