**CPSC 230: Computer Science I**

**Spring 2018**

**Programming Assignment 8**

**Due: May 10, 2018 @ 11:59pm**

**Overview**

In this assignment, you will get practice with files by implementing a simple encryption algorithm for a text file.

One of the most basic encryption algorithms possible is to simply replace a letter in the alphabet with another letter. The only constraint is that this replacement must be one-to-one: every letter can only be replaced by the same letter throughout the text.

Suppose you have a file, replace.txt, that specifies the replacement for every lowercase and uppercase letter in the alphabet. This file consists of 52 rows (26 for lowercase, 26 for uppercase) of the following form:

X Y

This is interpreted to mean X is replaced by Y. So basically replace.txt is just providing a mapping of characters in the unencrypted text to characters in the encrypted text. (At this point you should be thinking about dictionaries…)

Your job is to write two functions to encrypt and decrypt a file using the scheme above.

The first function, encrypt, will take the name of an unencrypted file and your replacement/mapping file as parameters. You should read the unencrypted file and apply the mapping. After encrypting, your function should write the encrypted text to another file, encrypted.txt

The second function, decrypt, will take the name of an encrypted file as a parameter, along with the same replace/mapping file used to encrypt it. You will read the encrypted file, reverse the mapping, and output the result to unencrypted.txt.

To make things easier, you need not worry about encrypting or unencrypting any characters not in the mapping file (eg. punctuation). Punctuation should however remain in all files. Your functions need not return anything since they are writing to files.

**Due Date**

This assignment is due at 11:59 pm on 05-10-2018. Submit via Blackboard. It should be labeled firstinitiallastname\_Assignment8. Please make sure to include all the required files (README, source files).

**Grading**

Your program will be evaluated for correctness and elegance. In particular, you should make sure your code is properly commented and obeys standard naming conventions.