Kaitlyn Chau
Professor Rahmati
CSE 331
Homework 1
9/19/24

Task 1:
After taking in plaintext and cipher key as input, I loop through the entire plaintext and ignore the non alphabet characters. For the alpha characters, I find the ASCII characters associated with them so that I can apply the number shift based off of the current char of the key in rotation. The uppercase or lowercase of the letter matters because the ASCII will be at different places. After accounting for the ASCII of upper or lower to put it to 0 temporarily to add the key placement, add the ASCII starting for upper or lower back in to get the letter. After applying the key to only to the alpha chars and repeating the key as needed, the ciphertext result is encoded and printed to the terminal.

Task 2:
After taking in the ciphertext and cipherkey as input, similarly loop through the entire plaintext as mentioned in task 1. For the alphabetic characters, instead of adding the key, we are subtracting the key from the ASCII since we are doing the reverse of the encoding process. Wrap around is accounted for by taking mod. And after returning the ASCII for the starting place of upper or lower, we have found the plaintext which is printed in the terminal.

Task 3:
I looked into reference [1] for the expected frequency of each alphabet letter. Inputs ciphertext and cipher key length are taken in from the terminal. To simplify the process, only consider the alphabet letters in the ciphertext and make them all into uppercase letters so handling the ascii is the same. The ciphertext is split into an array with each entry being all the characters that were shifted by each cipher key letter. For each entry/column in the array, we do frequency analysis by perform chi squared and use the list from reference [1] to get the expected values. For the shifts that result in the least chi squared value, we take that as the best shift and add that to the cipherkey. Then use code from task2 to decode and print the requested values into the terminal.

Task 4:
Attempted to brute force the length of the cipher key. Used my code from task 3 for the rest.

How to run the code:
There are bash shell scripts named to the hw naming convention that can run by calling it like ./task1.sh when in the current directory. After running the script that runs the code, the user is prompted for inputs but without a prompt that says so. Results will print in the terminal.

References:

[1] https://en.wikipedia.org/wiki/Letter_frequency