Code Project 1 - Encryption

By: Owen Hillary

Resources:

For the most part I used no outside help or resources. It is possible I had to verify syntax or paramatier of a built-in function throught an online form, but I believe I did not have to for this project.

The Process:

During my journey through out this project I found my self at three stages:

Stage One: In this stage Is where a did almost all of my coding, I found that I did not really struggle with any why writing the code. However there was one thing that stumped me for a little tiny bit, this was during the Xor chiper, when converting the binary back to base ten, I using a base ten interpretation of the binary string, instead of a base two interpretation. This lead me to also find out that I had written code that does the Xor function when I could of just used a built in function :(

Stage Two: In this stage I wrote out all my unit test, I was originally going to do this as I worte the code, but I was not sure how to express some of the outputs of the functions. This section was kinda of drag to finish, just because I had already felt I had finish when my code was done. For each class I test a couple valid functions, and this I have a list of invalid inputs, I run this list for every class, ensuring a type error is raised.

Stage Three: This was my final step of the process, where I had to write this journal, this was 100% the hardest step I faced. Not because writing is difficult, but its not very fun, where as I really enjoy coding. This ontop of the fact I already felt accomplished and finished with my work after the coding portiaion made this hard. Ultimately I ended up doing this very last minute (Realtivly), I had most of my code done with a few days of receiving the project, and I writing this with a few days before it is due.

Duplicate Code:

Every Class that I have written uses lots of duplicate code, one that they all use is the type verification code. To generilze this I created the function called valid_type, this function takes a list of inputs, and a single type and checks to see if all elments in the list are of the same type as the entered type. If any of the elements are not of the desired type a type error is raised. Additionally many of my classes have sections of duplicate code when it comes to their respective Enyctipon and Decyption process. In order to avoid repeating these longs line of code, I made functions to hold most classes process of encryption/decyption. Some of these functions had similar aspects to them, for example a key that loops to match the length of the text, or having to scale alphabetical letters by a set integer, so I made several functions for these class functions to share.

The following is a list of all the functions names I have to avoid repetitive code -

- find_shift(text: str) -> int:
    - Find the shift associated with a letter for Vigenere Cipher

- looping_alphabet(letter: str, change: int) -> str:
    - Shifts a letter based on the changes value, loops back to A or Z when value is shift passed them.

- looping_key(text: str, key: str) -> str:
    - Loops input key to match length of text string

- owens_reveres(input_str: str) -> str:
    - Returns a string that is reversed

- return_key(value: str, dic: dict) -> str:
    - Return what key value is associated with an item, in a dictionary where every item has unique key
    - (This class is not really needed as I found a better way to do what I need it to do, but if its aint broke dont fix it)

- the_caesar_cipher(text: str, key: int) -> str:

    - Calls looping_alphabet for each letter in text by value key and returns a whole string

- the_vigenere_cipher(text: str, key: str, mod: int = 1) -> str:
    - Calls looping_alphabet for each letter in text by the find_shift value of key value and returns a whole string

- the_xor_cipher(text: str, key: str) -> str:
    - X or's the binary values of each letter of Text and String, providing a new binary value, that is then convert to a character, that is then added to a string
    -
- valid_type(var: list, val_type):
    - Takes multiple input values and raises error if any of them are not instance of val_type