Assignment One: Implementation of Monoalphabetic Cipher

CE235 Assignment One

2020-2021

University of Essex

1. Task of Assignment One

The aim of Assignment One is to write a Python program, which will implement the monoalphabetic cipher. The introduction of the monoalphabetic cipher can be found from the Week 1 lecture notes.

Specifically, this assignment task is expected to do the following:

- **Design and develop a Python program with encryption and decryption functions for monoalphabetic cipher.**

A sample Python program caesar.py for caesar cipher is provided for reference. The sample Python program can be run with default demo message and key from the command line in Terminal like this:

```
python caesar_ciphyer.py
```

This reference program caesar_ciphyer.py encrypts a given message and then performs decryption. After the caesar_ciphyer.py program is run, it will display the plaintext, ciphertext, and the decrypted plaintext. If the cipher works correctly, the plain text and decrypted text should be the same.

Your program for monoalphabetic cipher should be called something like mc_registrationnumber.py (see below). Your program must run from the command line like this:

```
python mc_registrationnumber.py
```

The outputs of mc_registrationnumber.py (including exactly the key, plaintext, ciphertext and decrypted text) are required to be displayed, following the display format given in the reference program caesar_ciphyer.py.

2. Hint

For the monoalphabetic cipher encryption, you should find the index of a symbol in the plain alphabet LETTERS, and the corresponding symbol with the same index in the key for the cipher. The decryption process follows similar idea. The Python function find() can be used to find the index of an element in a string.
All the parts of the code you are expected to modify are highlighted in the Appendix A of the sample program for caesar cipher. You should not change the other part of the reference program.

3. How to submit

Submit one python program file to Faser called:

```
mc_registrationnumber.py
```

For example, if your registration number is 123456, your filename will be: mc_123456.py

Submission Deadline: Friday, Week 19, 12-Feb-2021, 11:59:59

4. Marking Scheme

There is 5% of the overall module marks for this assignment. We will test your program, running it from the command line. If it does not work from the command line, you will not get the mark. If it works correctly as expected, you will get the full 5% marks. If it can be run from the command line but does not produce expected results, you may get some partial marks, depending on how close the results are to the specification.

5. Plagiarism

You should work individually on this project. Anything you submit is assumed to be entirely your own work. The usual Essex policy on plagiarism applies: http://www.essex.ac.uk/plagiarism/.

Appendix A. Sample Python Program for Caesar Cipher

```
import sys

#-------------------------------------------------------------------------
### Note: Allow the program to be run from the command line:

##  You can simply use the default message and key given in the program
#       python caesar_cipher.py

##  You can also use message and key given in command line in Terminal (not required),
##  where, atestmessage is the message (no space!), 4 is the key (an integer for Caesar cipher!)
#       python caesar_cipher.py atestmessage 4

# the alphabet with all symbols in the set can be encrypted
LETTERS = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'


def caesar_cipher(message, mode, key):
    # stores the encrypted/decrypted form of the message
    ciphertext = ''

    # run the encryption/decryption code on each symbol in the message string
    for symbol in message:
        if symbol in LETTERS:
            # get the encrypted (or decrypted) index of this symbol in the LETTERS
            num = LETTERS.find(symbol) # index of the uncoded symbol in LETTERS
            if mode == 'encrypt':
                num = num + key
            elif mode == 'decrypt':
                num = num - key
            else:
                print('Correct operation mode is needed')
                exit()

            # handle the wrap-around if num is larger than the length of LETTERS or less than 0
            num = num % len(LETTERS)

            # add encrypted/decrypted number's symbol at the end of translated
            ciphertext = ciphertext + LETTERS[num]

        else:
            # just add the symbol without encrypting/decrypting if it is not in the set LETTERS
            ciphertext= ciphertext + symbol

    # print the encrypted/decrypted string to the screen
    return ciphertext


def monoalphabetic_cipher(message, mode, key):
    # The body of this function is to be completed by you in your own python program

    return ciphertext



if __name__ == '__main__':

    # Determine the number of arguements in the command line
    numArgv = len(sys.argv)

    # the default string to be encrypted/decrypted
    message = a secret message.'

    # the default key used by the caesar cipher
    key = 3
    ##  Note that for the monoalphebatic cipher you should use the following default key
    ##  by uncommenting the following statement
    # key = 'jklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghi'

    if numArgv == 2:
        # get the message from the input given in the command line
```

You need implement the main function for your cipher

You need uncomment this statement to use the key for your cipher

```
    message = sys.argv[1]
elif numArgv == 3:
    # get the message and key specified the input given in the command line
    message = sys.argv[1]
    key = int(sys.argv[2])

elif numArgv > 3:
    # Instruction on providing message and key from command line
    print('+++Please input the exected message and key with correct format')
    print('+++python caesar_cipher.py my_message key')
    print('+++where no space in my_message, key needs to be an integer for caesar cipher')

    print('+++For example: ')
    print('+++python caesar_cipher.py my_secrete_message 3')

    exit()


# tells the program to encrypt or decrypt
mode = 'encrypt' # set to 'encrypt' or 'decrypt'
ciphertext = caesar_cipher(message, mode, key)

mode = 'decrypt'
decrypttext = caesar_cipher(ciphertext, mode, key)


### Note: don't change the following code in your own program for displaying program outputs!!!
print('##################################################')
print('Cipher with key: ', key)
print('##################################################')
print('Plain message: ', message)
print('Ciphertext: ', ciphertext)
print('Decrypted text: ', decrypttext)
```

You should change them to the function names for your cipher

Don't change this part of code in your own program