Index

1.Introduction

The purpose of this lab report is to answer the Lab Assignment questions given by Dr Hossein Anisi on MOODLE. [1]
Python 3.6.2 is used as the programming language.

2.Question 1

1. Decrypt the following message in python.

HUKHZ MVYCH SVYAO HAJHU UVAIL JVTWB ALKIF ZAHAB YLOLO
HZWHZ ZLKAO YVBNO TVYLI HAASL ZHUKW LYPSZ AOHUF VBOHC
LPUNV SKAOV BNOFV BILAD PJLOP ZOLPN OAHUK OLJVT LZUVD MYVTA
OLZAV YTPUN VMPZL UNHYK VMDOP JODLI LHYAP KPUNZ HUKNY
LHADL HYPUL ZZPZV UOPTV YPDVB SKDHR LOPTO PZUHT LPZWL YLNYP
UMYVT SVYKV MAOLY PUNZI VVRMP CLJOH WALYV UL

Your program should decrypt the above ciphertext and display the plaintext.

Answer 1

As a solution to this question, the brute force method was preferred to frequency analysis. The very first reason for this preference is that the given text is too short and does not give the desired data to the user with frequency analysis method. Because, letter frequency in the text could be very different from standard English letter frequency.

Task 1

```
encrypted_text= """HUKHZ MVYCH SVYAO HAJHU UVAIL JVTWB ALKIF ZAHAB YLOLO
HZWHZ ZLKAO YVBNO TVYLI HAASL ZHUKW LYPSZ AOHUF VBOHC
LPUNV SKAOV BNOFV BILAD PJLOP ZOLPN OAHUK OLJVT LZUVD MYVTA
OLZAV YTPUN VMPZL UNHYK VMDOP JODLI LHYAP KPUNZ HUKNY
LHADL HYPUL ZZPZV UOPTV YPDVB SKDHR LOPTO PZUHT LPZWL YLNYP
UMYVT SVYKV MAOLY PUNZI VVRMP CLJOH WALYV UL"""
encrypted_text=encrypted_text.lower()
encrypted_text=encrypted_text.replace(" " , "")
```

In task 1, a variable (encrypted_text) is created for holding the encrypted text.
The entire text is converted to lowercase.
All space characters in the text have been deleted.

Task 2

```
shifting_number=1
while shifting_number<=26 :
    plain_text=""
    for counter in encrypted_text:
        if (ord(counter)+shifting_number)>122:
            plain_text=plain_text + chr(ord(counter) - 26 +
shifting_number)
        else:
            plain_text=plain_text + chr(ord(counter) +s hifting_number)
    shifting_number = shifting_number + 1

    print ( "shifting number", shifting_number)
    print ("your deencryped message is",plain_text )
    print ( )
```

In task 2, two loops are created, a *while* loop and a for loop inside it. For loop is separating the text to letters, then converting the letters into numeric equivalents (via ord( ) function). After that numeric equivalents are summed with the (shifting_number). In this way, a new letter is created according to Caesar cipher (via chr( ) function). While loop makes this work for every letter (a-z).

Output 1

As can be seen from the picture, Figure 1 , the program can produce meaningful output when shifting number equal to 20.

Figure 2 output of Lab_Assignment_1_1.py

2. Question 2

2. Encrypt a message with Vigenere Cipher in python.

Your program should read a plaintext and a key, encrypt the plaintext with the key and display the ciphertext.

Answer 2

As a solution to this question, although there is an easier way, using modular arithmetic [2], the hard way is chosen by creating an array of 676 members. The purpose of selecting this method is also to present the Vigenere table visually at the same time.

Task 1

```
#################### TASK 1 ##########################################
alphabe="abcdefghijklmnopqrstuvwxyz"
alphabe2="*abcdefghijklmnopqrstuvwxyz"
print(*alphabe2,sep=" | ")
vigenere_array = ""
for i in range (0,26):
    tablo=""
    tablo= alphabe[i]
    for j in range (i, i + 26):
        vigenere_array = vigenere_array + alphabe[ j % 26]
        tablo = tablo + alphabe[j % 26]
    print(*tablo,sep=" | ")
```

In task 1, a variable (alphabe) is created for holding letters in alphabet. After that, an empty string variable is (vigenere_array) created for holding whole Vigenere table letters. With loop nest containing two for loop assign the letters to the string variable. Finally, the Vigenere table was printed on the screen

Output of Task 1

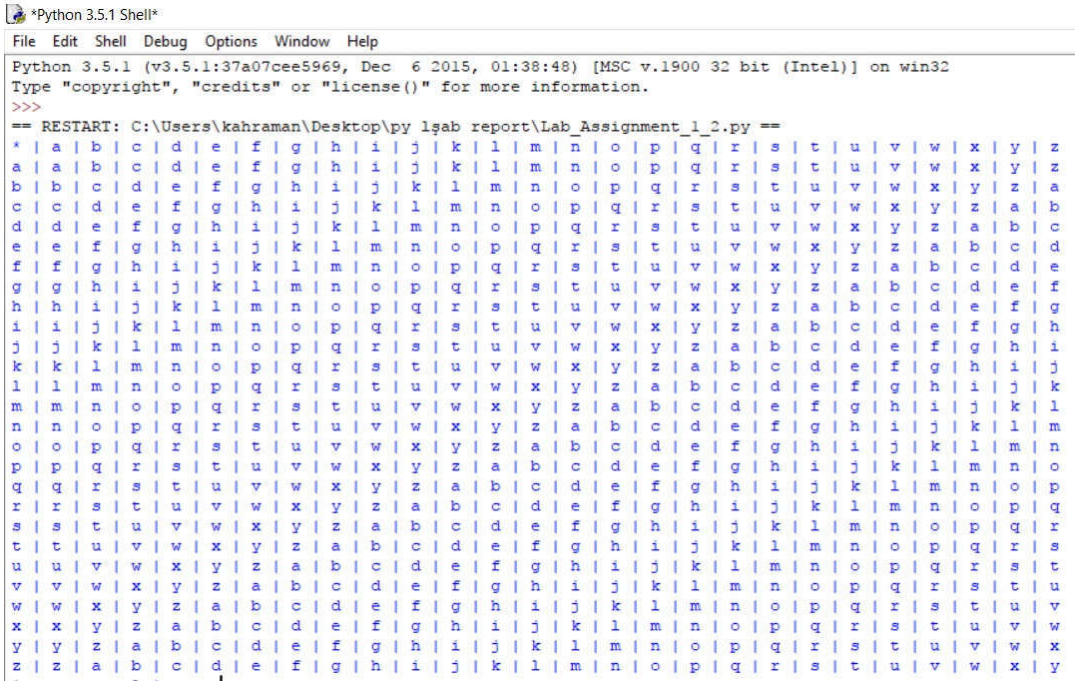

Figure 2 Output of Question 1, Task 1

Task 2

```
###################### TASK 2 ##############################################
plaintext=input("input your plain text")
plaintext=plaintext.lower()
key=input ("input your key")
key=key.lower()
key_len=len(key)
```

In task 2, two variables (plaintext and key) are created for holding plain text and key.
Also requested from the user for inputting values these two variables.

Task 3

```
###################### TASK 3 ##############################################
for i in range (key_len,len(plaintext)):
        key=key+key[i%key_len]
```

In task 3, the key value is being extended to bring the same length with plain text via for loop

Task 4

```
###################### TASK 4 ############################################
cipher_text=""
for i in range (0,len(plaintext)):
    cipher_text_index= (ord(plaintext[i])-97)+(ord(key[i])-97)*26
    cipher_text=cipher_text + chr(ord(vigenere_array[cipher_text_index]))
print (cipher_text)
```

In task 3, by subtracting the numerical value of "a" (97) , every the plain text letter is used as a raw and the every key letter used as a column in the the Vigenere table. When a key letter value is multiplied by 26 and summed with plain text letter value, the number reached is an index number cipher text letter. For loop makes this work for every letter in the plain text and these letters are stored in an empty string variable (cipher_text).
Finally, the result (ciphered text) is printed on the screen

Output 2

```
== RESTART: C:\Users\kahraman\Desktop\py lşab report\Lab_Assignment_1_2.py ==
* | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z
a | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z
b | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a
c | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b
d | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c
e | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d
f | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e
g | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f
h | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g
i | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h
j | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i
k | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j
l | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k
m | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l
n | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m
o | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n
p | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
q | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p
r | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q
s | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r
t | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s
u | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t
v | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u
w | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v
x | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w
y | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x
z | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y
input your plain textthere is a secret passage behind the picture frame
input your keyIHS
bowzlcqzcirkmjjmacxhkahymrtmoavkcbowswakamzlcnysul
>>>
```

Figure 3 output of Lab_Assignment_1_2.py