Cryptography and Network Security Lab

Assignment 3
Implementation and Understanding of Vigenere Cipher

2019BTECS00058
Devang K
Batch: B2

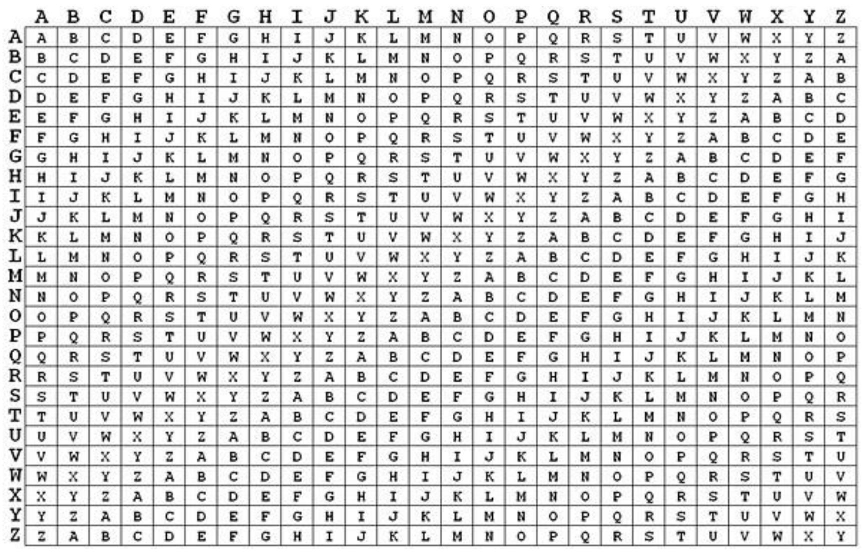
Title: Implementation and Understanding of Vigenere Cipher

Aim: To Study, Implement and Demonstrate the Vigenere Cipher Algorithm

Theory:

Vigenere Cipher is an encryption method that is a form of poly-alphabetic substitution. It uses the Vigenere matrix to perform the encryption and decryption.

A Vigenere matrix is a table where alphabets are written in 26 possible rows that are Caesar Ciphered with increasing the key. This table alongside the key character and the character to encrypt/decrypt, gives us the answer.



	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

Illustration:

Encryption

Say we wish to encrypt – ‘ATTACKATONCE’ and say with a key of ‘DEVANG’. Firstly, we generate a cycle-repeating key of the size of the plaintext. So, length of ‘ATTACKATONCE’ = 12. Therefore, the Cyclic Repeating Key would be ‘DEVANGDEVANG’.

Then, we now go character by character and for each character, we take the row element of the plaintext character and the column element of key character. The corresponding element is our Cipher-text.

Therefore, traversing our plaintext, we get row(‘A’) and col(‘D’) i.e. (0,3) which would be ‘A’. Similarly next, row(‘T’) and col(‘E’) i.e. (19, 4) which is ‘X’. Thus, by the end, we would get the generated Ciphertext of ‘DXOAPQDXJNPK’.

Decryption

Say we wish to decrypt our Ciphertext of ‘DXOAPQDXJNPK’ with the key of ‘DEVANG’. We shall first generate the cycle-repeating key, we get ‘DEVANGDEVANG’. Then we shall now traverse every character of the Ciphertext.

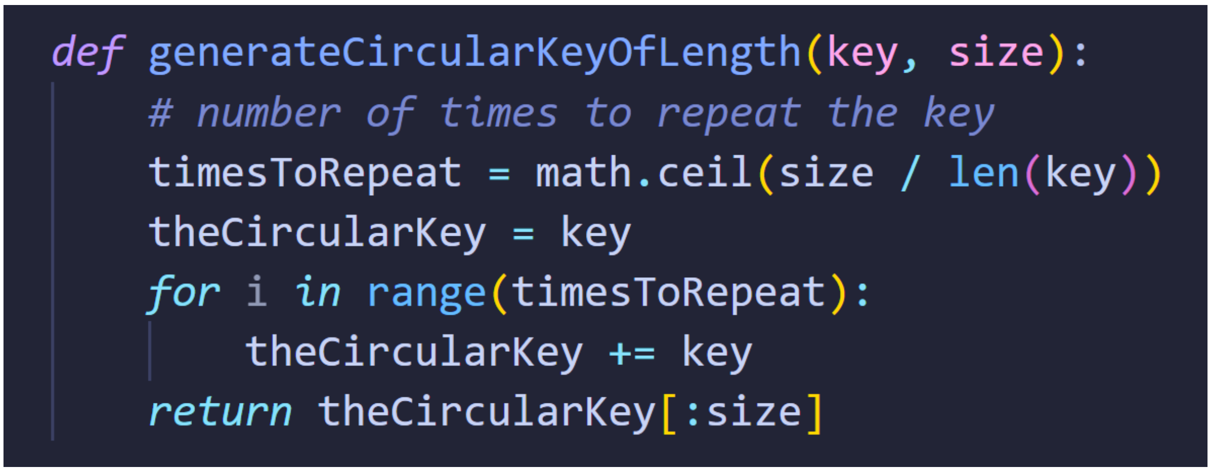
Starting with cipher-character ‘D’ and key-character ‘D’, we take the row of the table corresponding to the key-character and then find the column whose value is the cipher-character ‘D’. Therefore, we get row(‘D’) – 3 and the column corresponding to ‘D’ in the 3rd row as 0. Hence, the value shall become (3, 0) – ‘A’. Similarly, next is key character ‘E’ and cipher character ‘X’. We look at row of ‘E’ – 4 and column corresponding to ‘X’ in ‘E’ – ‘T’. Thus, we continue to traverse till the end until we generate the plaintext ‘ATTACKATONCE’.

We illustrate more examples using the code.

Code:

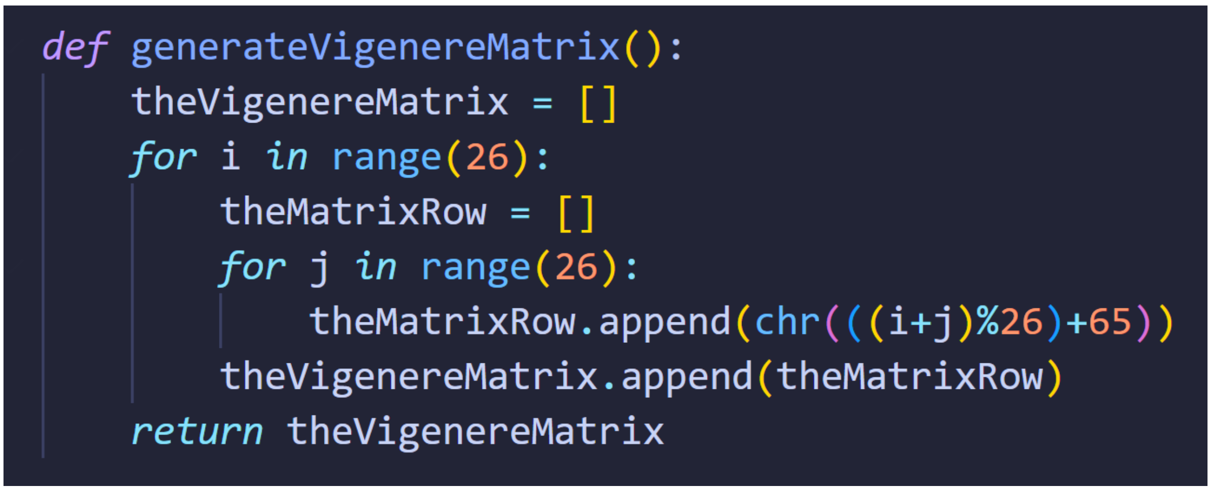
Some important functions we need for the code, common for encryption and decryption are:

A function to generate the circular key of given length of plain/cipher-text.



```python
def generateCircularKeyOfLength(key, size):
    # number of times to repeat the key
    timesToRepeat = math.ceil(size / len(key))
    theCircularKey = key
    for i in range(timesToRepeat):
        theCircularKey += key
    return theCircularKey[:size]
```

A function to generate Vigenere Matrix.



```python
def generateVigenereMatrix():
    theVigenereMatrix = []
    for i in range(26):
        theMatrixRow = []
        for j in range(26):
            theMatrixRow.append(chr(((i+j)%26)+65))
        theVigenereMatrix.append(theMatrixRow)
    return theVigenereMatrix
```

Then corresponding to these, we would have the functions to traverse each character with key and compare with the Vigenere Table and generate the Ciphertext or to generate the Plaintext.

Code for Encryption:

```python
import math

def printMatrix(fenceMatrix):
    for i in range(len(fenceMatrix)):
        for j in range(len(fenceMatrix[0])):
            print(fenceMatrix[i][j], end=' ')
        print()

def generateVigenereMatrix():
    theVigenereMatrix = []
    for i in range(26):
        theMatrixRow = []
        for j in range(26):
            theMatrixRow.append(chr(((i+j)%26)+65))
        theVigenereMatrix.append(theMatrixRow)
    return theVigenereMatrix

def generateCircularKeyOfLength(key, size):
    # number of times to repeat the key
    timesToRepeat = math.ceil(size / len(key))
    theCircularKey = key
    for i in range(timesToRepeat):
        theCircularKey += key
    return theCircularKey[:size]

vigenereMatrix = generateVigenereMatrix()

print("Vigenere Cipher")
print("Enter Plaintext: ", end='')
plaintext = input()
print("Enter the Key: ", end='')
key = input()

circularKey = generateCircularKeyOfLength(key, len(plaintext))

# To encrypt -> Row Index -> Plaintext, Col Index -> CipherText

encryptedText = ""

for i in range(len(plaintext)):
    rowIndex = ord(plaintext[i])-65
    colIndex = ord(circularKey[i])-65
    encryptedText += vigenereMatrix[rowIndex][colIndex]

print("\nCircular Key: ", end='')
print(circularKey)

print("\nVigenere Table:")
printMatrix(vigenereMatrix)

print()
print()

print("Plaintext:", plaintext)
print("Encrypted Text", encryptedText)
```

Code for Decryption:

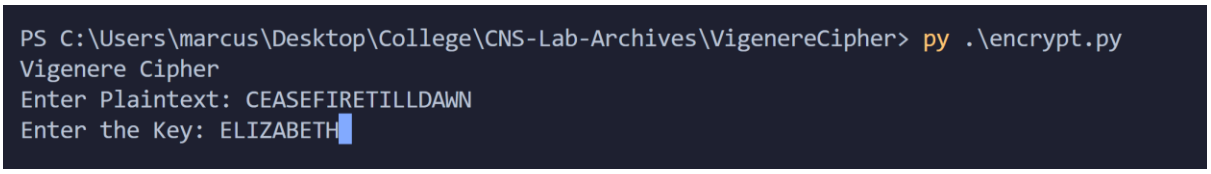
```
import math

def printMatrix(fenceMatrix):
    for i in range(len(fenceMatrix)):
        for j in range(len(fenceMatrix[0])):
            print(fenceMatrix[i][j], end=' ')
        print()

def generateVigenereMatrix():
    theVigenereMatrix = []
    for i in range(26):
        theMatrixRow = []
        for j in range(26):
            theMatrixRow.append(chr(((i+j)%26)+65))
        theVigenereMatrix.append(theMatrixRow)
    return theVigenereMatrix

def generateCircularKeyOfLength(key, size):
    # number of times to repeat the key
    timesToRepeat = math.ceil(size / len(key))
    theCircularKey = key
    for i in range(timesToRepeat):
        theCircularKey += key
    return theCircularKey[:size]

def getIndexFromRow(theRow, char):
    for i in range(len(theRow)):
        if(theRow[i] == char):
            return i

vigenereMatrix = generateVigenereMatrix()
```

```python
print("Vigenere Cipher")
print("Enter Encrypted Text: ", end='')
encryptedText = input()
print("Enter the Key: ", end='')
key = input()

circularKey = generateCircularKeyOfLength(key, len(encryptedText))

# To decrypt -> Row Index -> Key, Col Index -> Where the Enc Text Char is
found - that col Index correspondign char

plaintext = ""

for i in range(len(encryptedText)):
    rowIndex = ord(circularKey[i])-65
    plainTextValue = chr(getIndexFromRow(vigenereMatrix[rowIndex],
encryptedText[i])+65)
    plaintext += plainTextValue

print("\nCircular Key: ", end='')
print(circularKey)

print("\nVigenere Table:")
printMatrix(vigenereMatrix)

print()
print()

print("Encrypted Text", encryptedText)
print("Plaintext:", plaintext)
```

We now solve some examples with the code.
Say we wish to encrypt: ‘CEASEFIRETILLDAWN’

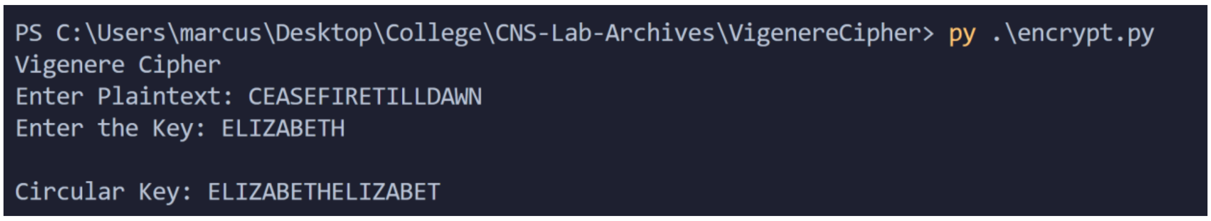
We choose the option to directly type the string.
And let’s take key of ‘ELIZABETH’.

We run our code for encryption:



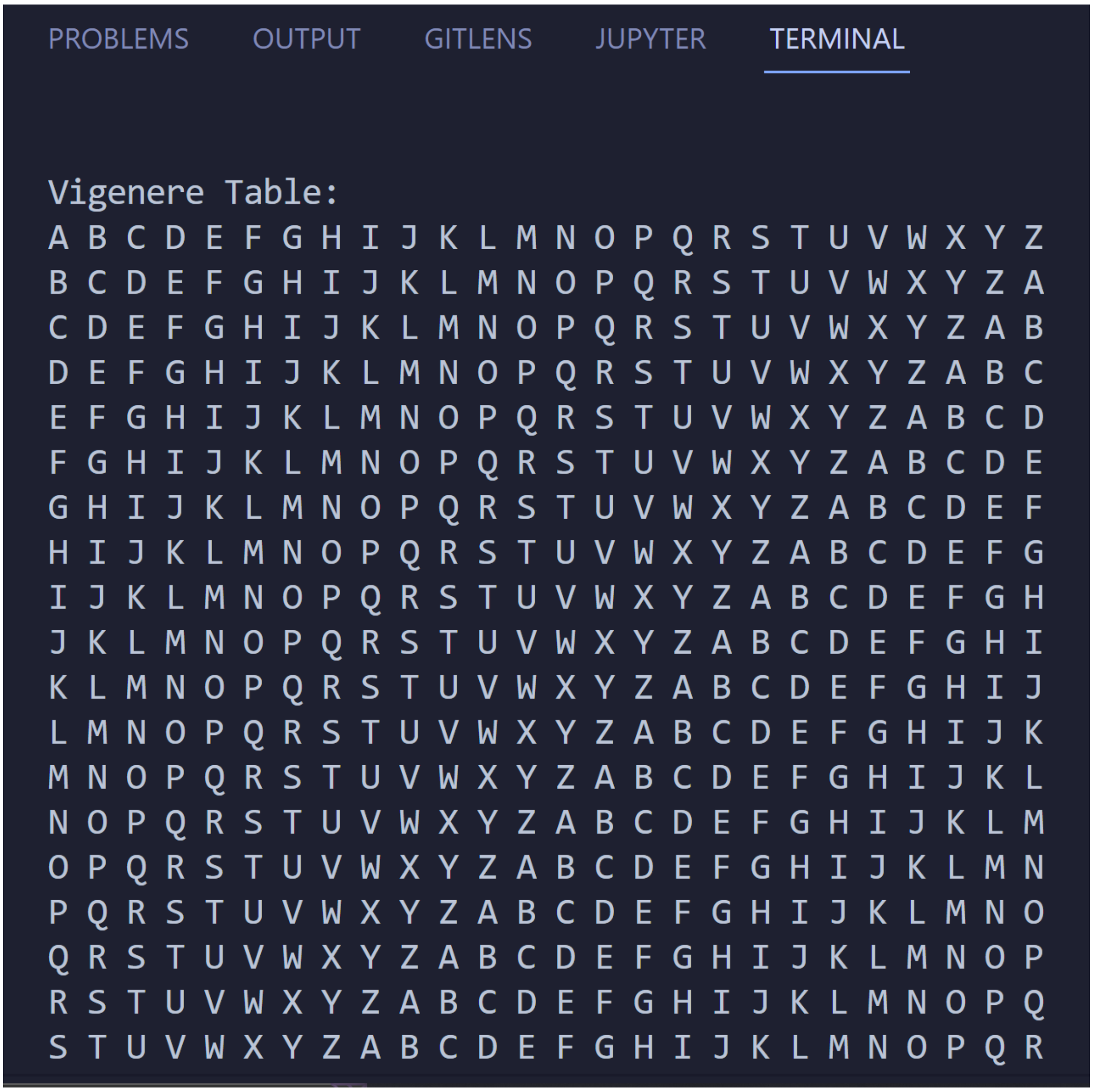
```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher> py .\encrypt.py
Vigenere Cipher
Enter Plaintext: CEASEFIRETILLDAWN
Enter the Key: ELIZABETH
```

It first generates the circular key:



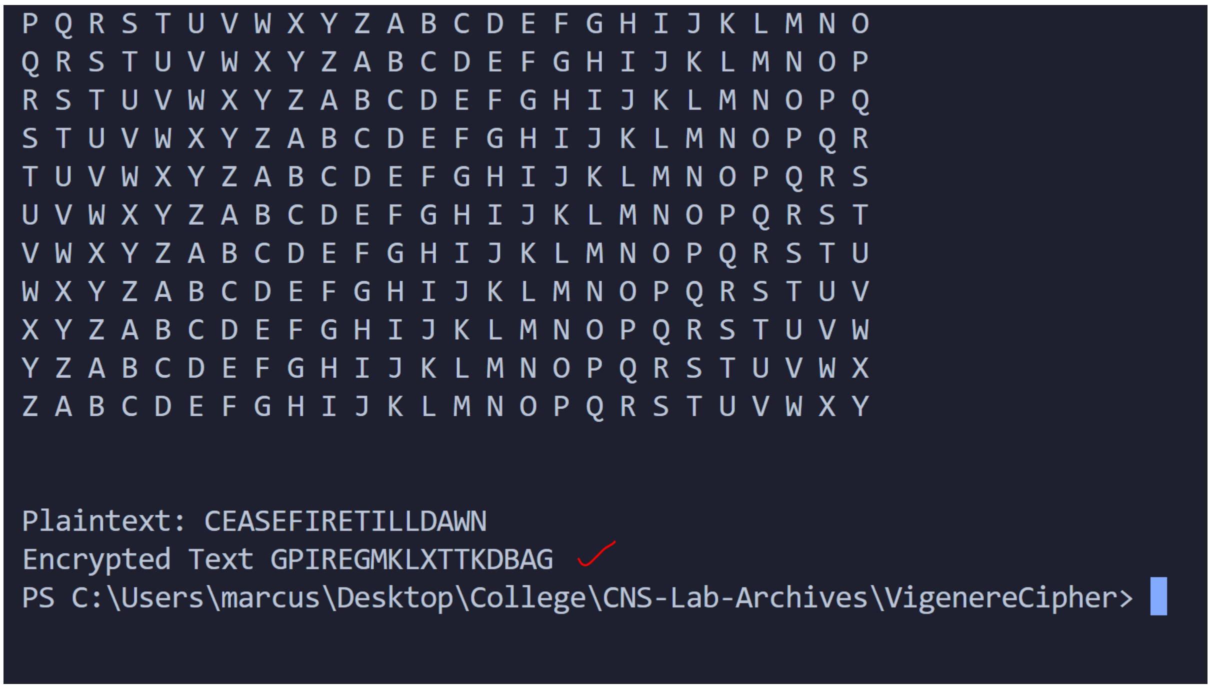
```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher> py .\encrypt.py
Vigenere Cipher
Enter Plaintext: CEASEFIRETILLDAWN
Enter the Key: ELIZABETH

Circular Key: ELIZABETHELIZABET
```

Then generates the Vigenere Matrix



```
Vigenere Table:
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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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
```

Then follows the algorithm to generate the cipher-text



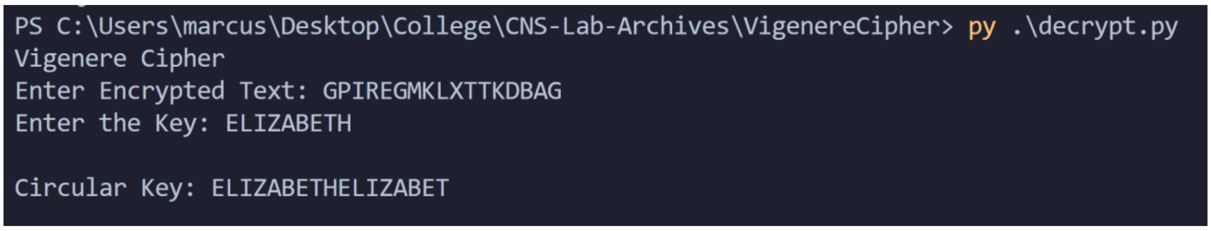
```
P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V 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 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 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 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 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


Plaintext: CEASEFIRETILLDAWN
Encrypted Text GPIREGMKLXTTKDBAG ✓
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher>
```

Thus, cipher-text of ‘CEASEFIRETILLDAWN’ with key ‘ELIZABETH’ is ‘GPIREGMKLXTTKDBAG’.

Now, we decrypt using the code.

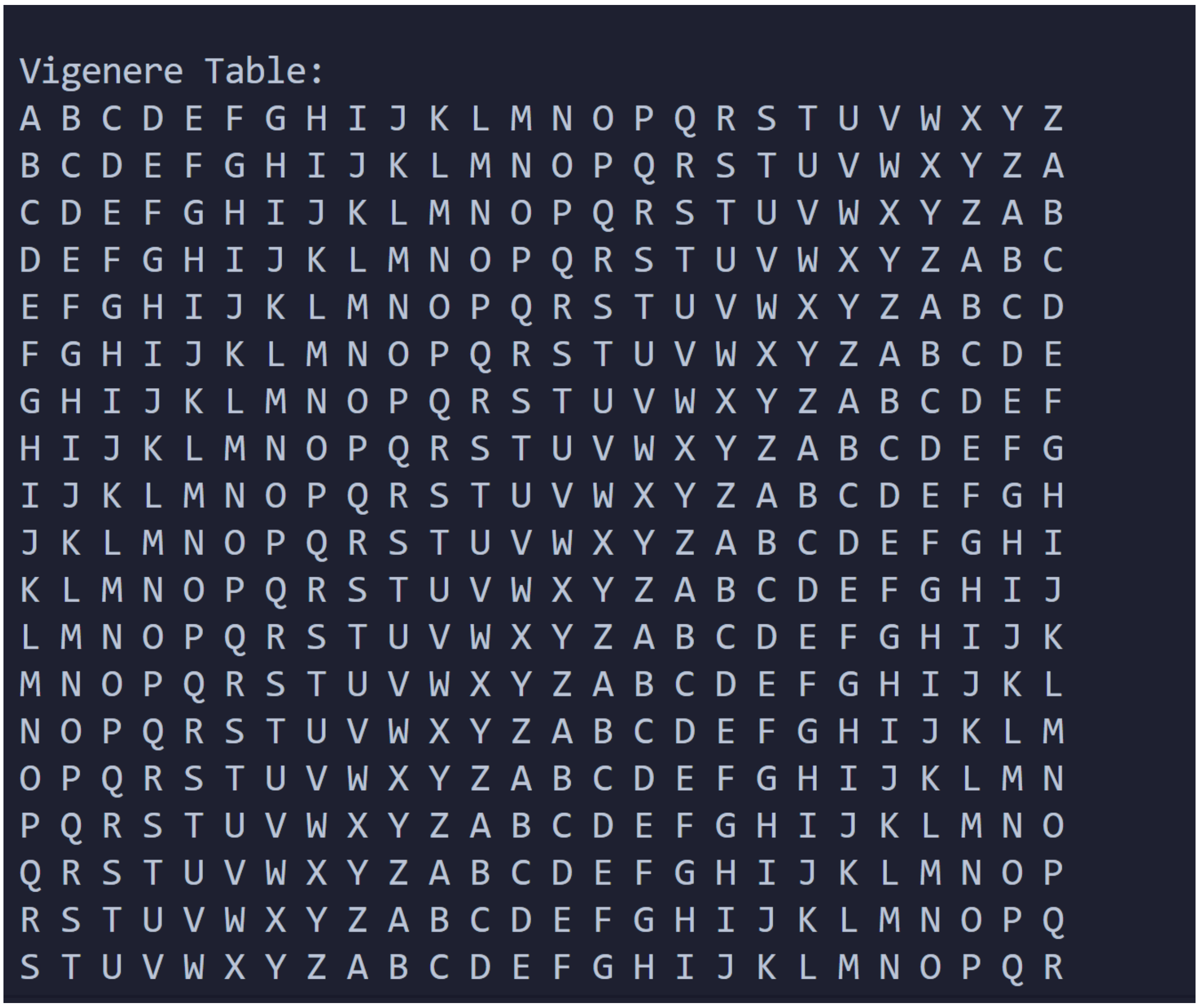
Our Cipher-text of ‘GPIREGMKLXTTKDBAG’ with key ‘ELIZABETH’:

It generates the Circular Key



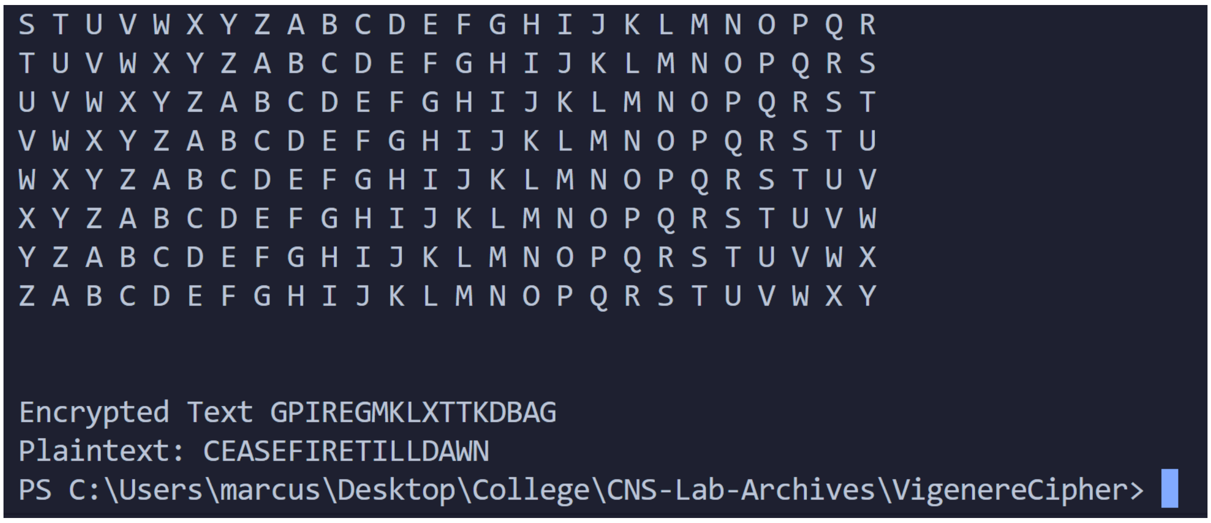
```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher> py .\decrypt.py
Vigenere Cipher
Enter Encrypted Text: GPIREGMKLXTTKDBAG
Enter the Key: ELIZABETH

Circular Key: ELIZABETHELIZABET
```

Then generates the Vigenere Table:



```
Vigenere Table:
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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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
```

And the decryption,



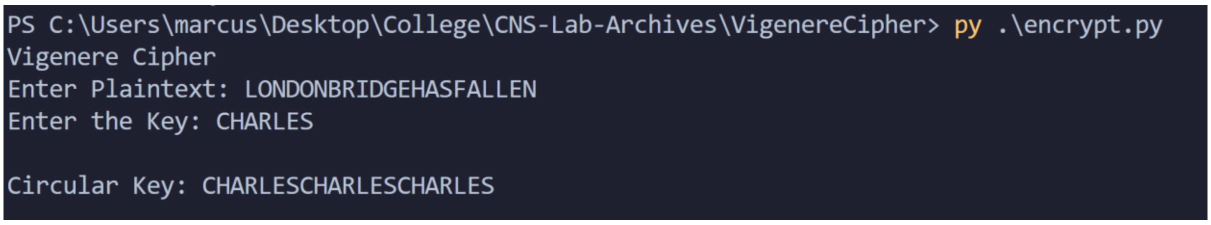
```
S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y 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 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


Encrypted Text GPIREGMKLXTTKDBAG
Plaintext: CEASEFIRETILLDAWN
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher>
```

Thus, the plaintext of cipher-text ‘GPIREGMKLXTTKDBAG’ with key ‘ELIZABETH’ is ‘CEASEFIRETILLDAWN’.

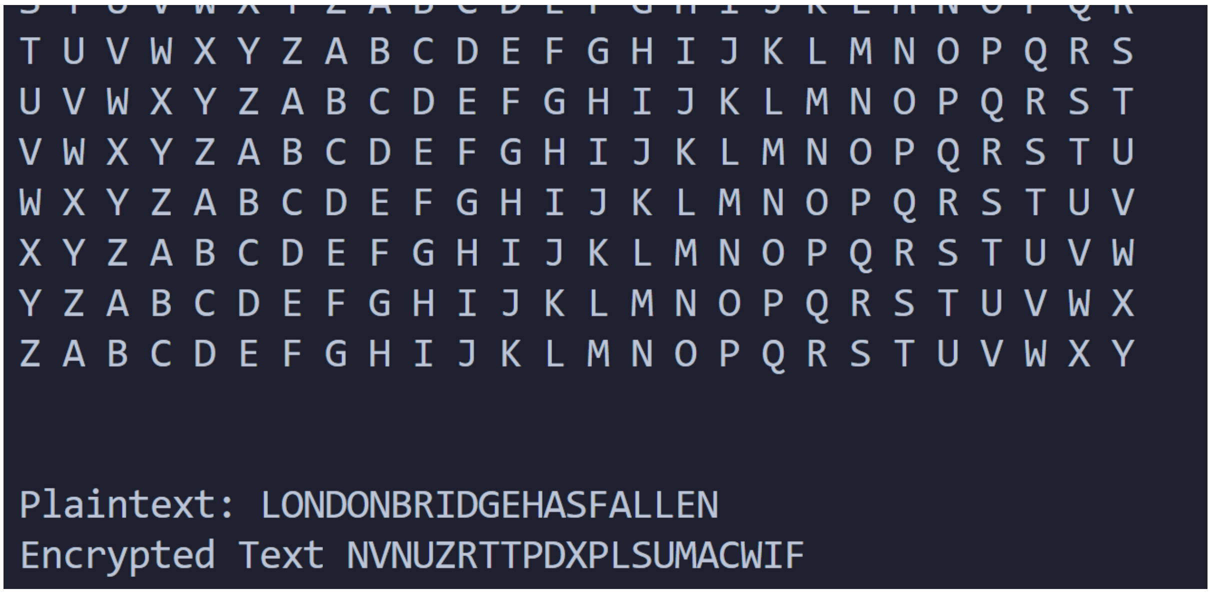
We look at a final example:

Say for ‘LONDONBRIDGEHASFALLEN’ with key ‘CHARLES’

We get:



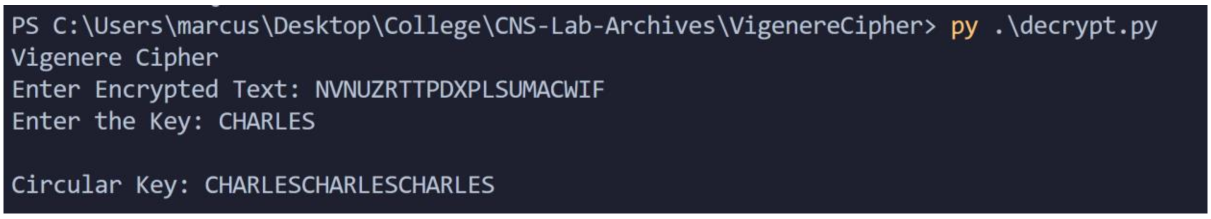
```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher> py .\encrypt.py
Vigenere Cipher
Enter Plaintext: LONDONBRIDGEHASFALLEN
Enter the Key: CHARLES

Circular Key: CHARLESCHARLESCHARLES
```



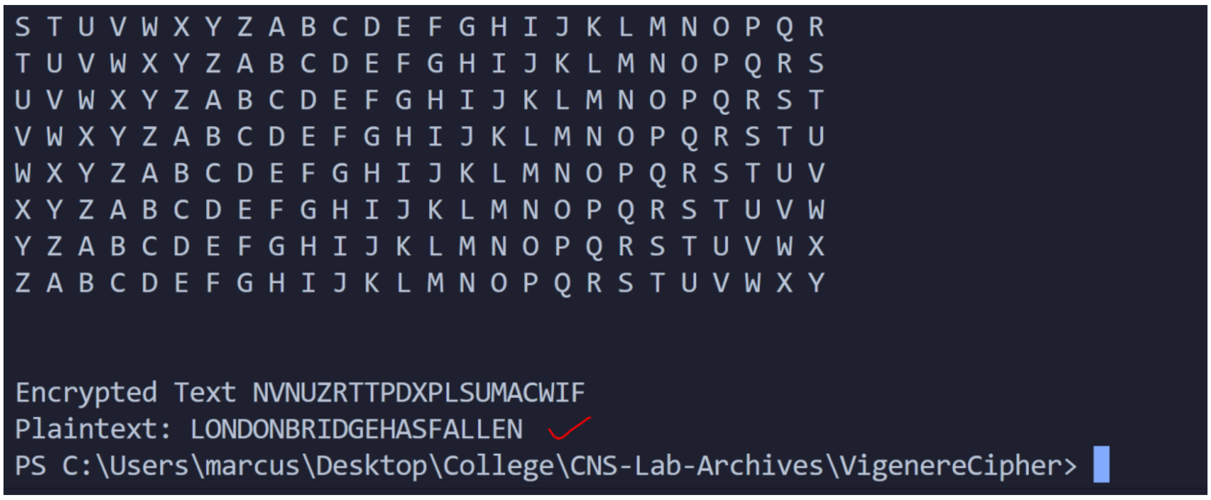
```
T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z 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


Plaintext: LONDONBRIDGEHASFALLEN
Encrypted Text NVNUZRTTPDXPLSUMACWIF
```

Thus, the encrypted text would be, ‘NVNUZRTTPDXPLSUMACWIF’.

We can decrypt it as:



```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher> py .\decrypt.py
Vigenere Cipher
Enter Encrypted Text: NVNUZRTTPDXPLSUMACWIF
Enter the Key: CHARLES

Circular Key: CHARLESCHARLESCHARLES
```



```
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 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 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 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 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 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 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 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


Encrypted Text NVNUZRTTPDXPLSUMACWIF
Plaintext: LONDONBRIDGEHASFALLEN ✓
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\VigenereCipher>
```

We get our plaintext back.

Thus, we demonstrated the working of the code with examples.

Conclusion:

Thus, the Vigenere Cipher algorithm was studied and demonstrated with the code. It is observed that Vigenere Cipher is much stronger and harder to crack than the Caesar Cipher as the key can be anything and not one of 26 values as in Caesar Cipher.