Experiment 04

Write a program in Java or Python to perform Cryptanalysis or decoding of Vigenere Cipher.

Roll No.	Manav Jawrani
Name	19
Class	D15-A
Subject	Security Lab
LO Mapped	LO1: To apply the knowledge of symmetric cryptography to implement classical ciphers.

Aim: Write a program in Java or Python to perform Cryptanalysis or decoding of Playfair and Vigenere cipher.

Introduction:

1. What is Cipher?

Ciphers, also called encryption algorithms, are systems for encrypting and decrypting data. A cipher converts the original message, called plaintext, into ciphertext using a key to determine how it is done.

2. What is Vigenere cipher?

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the *Vigenère square or Vigenère table*.

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

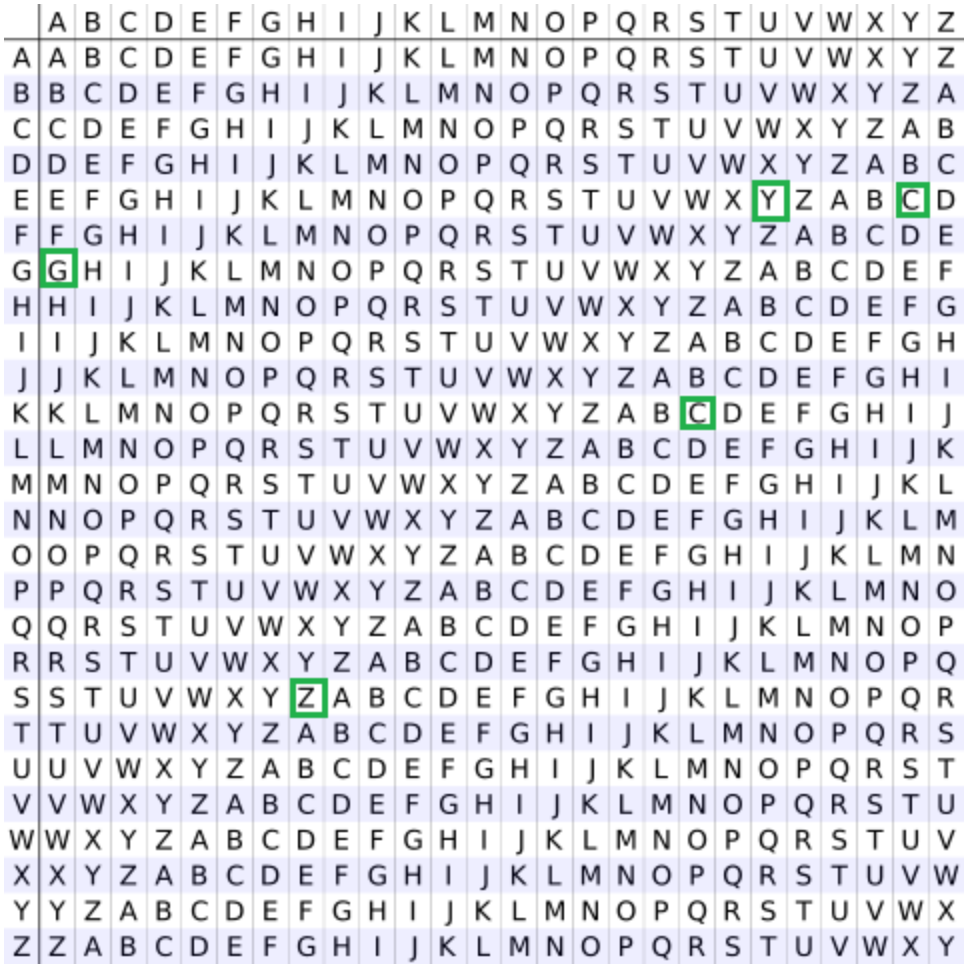
Encryption:

The first letter of the plaintext, G is paired with A, the first letter of the key. So use row G and column A of the Vigenère square, namely G. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E, and column Y is C. The rest of the plaintext is enciphered in a similar fashion.

Encryption

The plaintext(P) and key(K) are added modulo 26.

$E_i = (P_i + K_i) \bmod 26$



	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

Decryption:

Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext. For example, in row A (from AYUSH), the ciphertext G appears in column G, which is the first plaintext letter. Next, we go to row Y (from AYUSH), locate the ciphertext C which is found in column E, thus E is the second plaintext letter.

A more **easy implementation** could be to visualize Vigenère algebraically by converting [A-Z] into numbers [0–25].

Decryption

Di = (Ei - Ki + 26) mod 26

Algorithm:

Encryption:

The encryption can be described by the following formula:

$$C_i \equiv T_i + K_i \pmod{m}$$

Ci - i-th character of the ciphertext
Ti - i-th character of the open text
Ki - i-th character of the key phrase (if the key phrase is shorter than the open text, which is
usual, than the keyphrase is repeated to math the length of the open text)
m - length of the alphabet

Decryption:

The process of decryption is analogous. The key phrase is modularly subtracted from the ciphertext.

$$T_i \equiv C_i - K_i \pmod{m}$$

Ci - i-th character of the ciphertext
Ti - i-th character of the open text
Ki - i-th character of the key phrase (if the key phrase is shorter than the open text, which is
usual, than the keyphrase is repeated to math the length of the open text)
m - length of the alphabet

Code:

```
import java.util.Scanner;

public class VigenereCipher {
	public static void main(String[] args) {

		System.out.println("Vigenere Cipher\n");
		Scanner in = new Scanner(System.in);
```

```
		//	Allow the user to choose if they want to encrypt or decrypt a
message
		System.out.println("Press 1 to encrypt a message\nPress 2 to decrypt a
message");
		int input = in.nextInt();

		if (input == 1) {
			System.out.print("Enter the key in UPPER Case: \n");
			String key = in.next();
			System.out.print("Enter the message that would like to be
encrypted by Vigenere cipher: \n");
			String EMessage = in.next();
			String encryptMessage = encrypt(EMessage, key);
			System.out.println("The encrypted message is: \n" +
encryptMessage);
		}
		else if (input == 2) {
			System.out.print("Enter the key in UPPER Case: \n");
			String key = in.next();
			System.out.print("Enter the message that would like to be
decrypted by Vigenere cipher: \n");
			String DMessage = in.next();
			String decryptMessage = decrypt(DMessage, key);
			System.out.println("The decrypted message is: \n" +
decryptMessage);
		}
		else {
			System.out.println("Wrong Input!");
		}
		in.close();
	}

	//	Encryption
	//	Encryption Logic: Using ASCII Dec Representation:
	//	Example:
```

```
	//	ASCII: "H" is 72 && "S" is 83
	//	((72-65) + (83-65)) % 26 + 65 >> Encrypted "Z"
	public static String encrypt(String Message, String Key) {
		String EMessage = "";
		Message = Message.toUpperCase();
		for (int i = 0, j = 0; i < Message.length(); i++) {
			char letter = Message.charAt(i);
			EMessage += (char)(((letter - 65) + (Key.charAt(j)-65)) % 26 +
65);
			j = ++j % Key.length();
		}
		return EMessage;
	}

	//	Decryption
	//	Decryption Logic: Using ASCII Dec Representation:
	//	Example:
	//	ASCII: "Z" is 90 && "S" is 83
	//	(90-83+26) % 26 + 65 >> Encrypted "Z"
	public static String decrypt(String Message, String Key) {
		String DMessage = "";
		Message = Message.toUpperCase();
		for (int i = 0, j = 0; i < Message.length(); i++) {
			char letter = Message.charAt(i);
			DMessage += (char)((letter - Key.charAt(j) + 26) % 26 + 65);
			j = ++j % Key.length();
		}
		return DMessage;
	}
}
```

Output:

1. Encrypting the message:

```
java -cp /tmp/dbqa6IGRff VigenereCipher
Vigenere Cipher
Press 1 to encrypt a message
Press 2 to decrypt a message
1
Enter the key in UPPER Case:
OMKAR
Enter the message that would like to be encrypted by Vigenere cipher:
HiManav
The encrypted message is:
VUWAEOH
```

2. Decrypting the message:

```
java -cp /tmp/RVmuIB3DXE VigenereCipher
Vigenere Cipher
Press 1 to encrypt a message
Press 2 to decrypt a message
2
Enter the key in UPPER Case:
OMKAR

Enter the message that would like to be decrypted by Vigenere cipher:
VUWAEOH
The decrypted message is:
HIMANAV
```

Conclusion:

We have performed cryptanalysis and decoded the vigenere cipher using a java program.