NAME: PRERNA SUNIL JADHAV
SAP ID: 60004220127
BATCH: C2-2
BRANCH: COMPUTER ENGINEERING
COURSE: INFORMATION SECURITY LABORATORY
COURSE CODE: DJ19CEL603

EXPERIMENT 02

AIM: Study and Implement Vigenere Cipher.

THEORY: It is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is an cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the vigenere square or vigenere 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 alphabets, corresponding to the 26 possible Caesar ciphers.
A more easy implementation could be to visualize vigenère alphabetically by converting [A-Z] into numbers [0-25].

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

Decryption: $D_i = (E_i - K_i) \bmod 26$

Example:

The plaintext is : "PRERNA"

The key is : "BEST"

Plaintext	P	R	E	R	N	A
value	15	17	04	17	13	00
key	B	E	S	T	B	E
value	01	04	18	19	01	04
Encrypted	Q	V	W	K	O	E
value	16	21	22	10	14	04

The encrypted text is "QVWKOE"

CONCLUSION: The time complexity to convert the string into cipher text is O(n) where n is the length of the string. The space complexity is O(n).

Hence, we studied and implemented the vigenere cipher.

Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B. Tech (Computer Engineering)
Course:	Information Security Laboratory
Course Code:	DJ19CEL603
Experiment No.:	02

AIM: Study and Implement Vigenere Cipher.

CODE:

```
def generateKey(string, key):
    key = list(key)
    if len(string) == len(key):
        return(key)
    else:
        for i in range(len(string) -
                       len(key)):
            key.append(key[i % len(key)])
    return("" . join(key))

def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) +
             ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("" . join(cipher_text))

def originalText(cipher_text, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) -
             ord(key[i]) + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return("" . join(orig_text))

if __name__ == "__main__":

    string = input("Enter your message: ")
    keyword = input("Enter key: ")
    key = generateKey(string, keyword)
    cipher_text = cipherText(string,key)
```

```
    print("Ciphertext :", cipher_text)
    print("Original/Decrypted Text :",
        originalText(cipher_text, key))
```

OUTPUT:

```
PS C:\Users\Jadhav\Documents\BTech\Docs\6th Sem\IS\Code> & C:/msys64/mingw64/bin/python.exe "c:/Users/Jadhav/Documents
/BTech/Docs/6th Sem/IS/Code/Exp2/Vigenere.py"
Enter your message: HITHISISPRERNA
Enter key: VIGENERE
Ciphertext : CQZLVWZWKZKVAE
Original/Decrypted Text : HITHISISPRERNA
```