Name: Tushar Panchal

En.No: 21162101014

Sub: INS (INFORMATION SECURITY)

Branch: CBA

Batch:61

----------------------------PRACTICAL 06----------------------------

- ## AIM :

Alice wants to send some confidential information to Bob over a secure network.

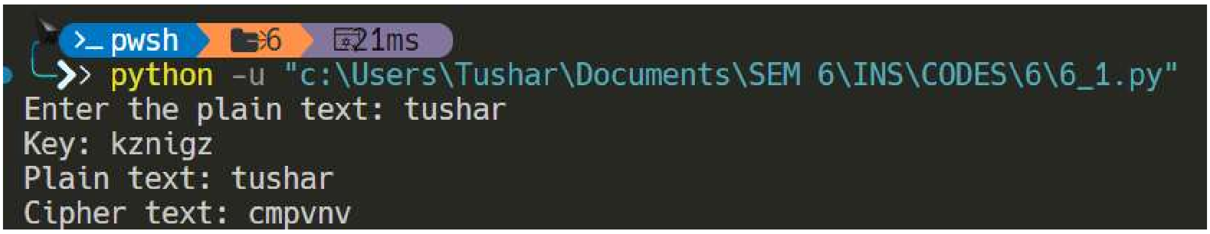
i) Create a system where the key will be generated randomly for encryption, and it will be changed with every message. Send three messages from sender to receiver and also decrypt the message at receiver end.

- ### Source Code :

```
import random

def generate_key(length):
    """Generates a random key of given length."""
    alphabet = "qwertyuiopasdfghjklzxcvbnm"
    return ''.join(random.choice(alphabet) for _ in range(length))

def encrypt(plain_text, key):
    """Encrypts the given plain text using the key."""
    alphabet = "qwertyuiopasdfghjklzxcvbnm"
    cipher_text = ""
    for i in range(len(plain_text)):
        plain_index = alphabet.find(plain_text[i])
        key_index = alphabet.find(key[i])
        cipher_index = (plain_index + key_index) % 26
        cipher_text += alphabet[cipher_index]
    return cipher_text
```

```

def main():
    plain_text = input("Enter the plain text: ").lower().replace(" ",
"")
    key = generate_key(len(plain_text))
    cipher_text = encrypt(plain_text, key)
    print("Key:", key)
    print("Plain text:", plain_text)
    print("Cipher text:", cipher_text)

if __name__ == "__main__":
    main()
```

✓ **Output :**

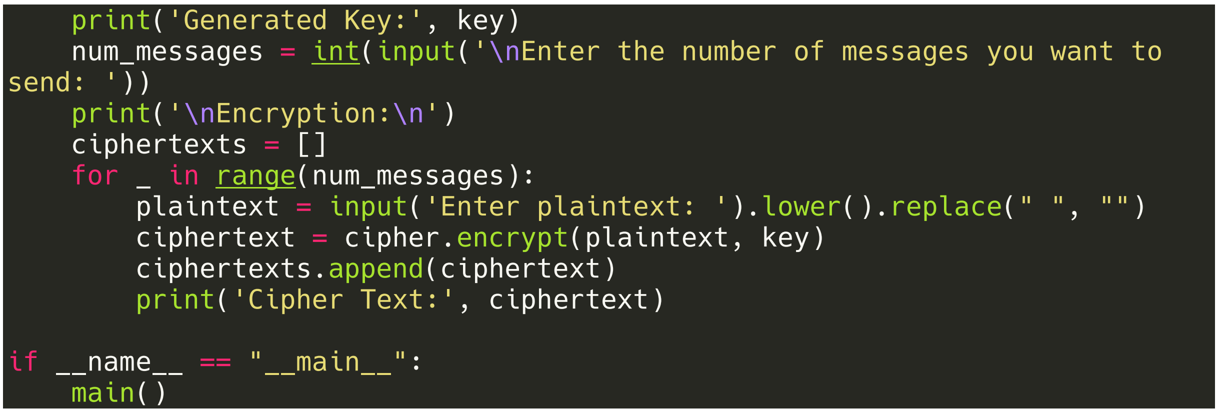


```
>_ pwsh  6  21ms
>> python -u "c:\Users\Tushar\Documents\SEM 6\INS\CODES\6\6_1.py"
Enter the plain text: tushar
Key: kznigz
Plain text: tushar
Cipher text: cmpvnv
```

ii) Provide encryption through vigener table as well. (Use Second Method)

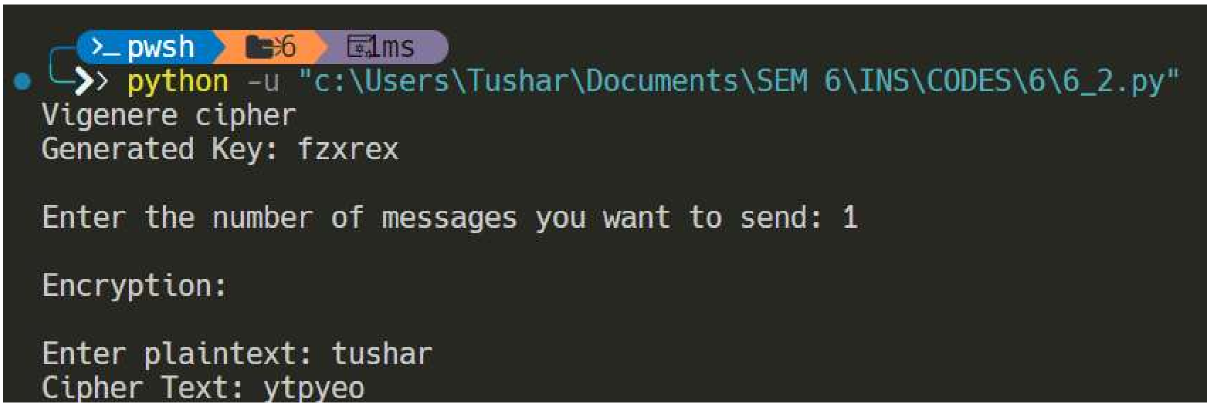
✓ **Source Code :**

```
import random as rd

class VigenereCipher:
    def __init__(self):
        self.alphabets = {chr(x + 97): x for x in range(26)}

    def generate_key(self, length):
        key = ''
        for _ in range(length):
            key += chr(97 + rd.randint(0, 25))
        return key

    def encrypt(self, plaintext, key):
        ciphertext = ''
        key_length = len(key)
        for i, char in enumerate(plaintext):
            if char.isalpha():
                shift = self.alphabets[key[i % key_length]]
                encrypted_char = chr(((self.alphabets[char] + shift) % 26)
+ 97)
                ciphertext += encrypted_char
        return ciphertext

def main():
    print('Vigenere cipher')
    cipher = VigenereCipher()
    key_length = rd.randint(6, 10)
    key = cipher.generate_key(key_length)
```



```python
    print('Generated Key:', key)
    num_messages = int(input('\nEnter the number of messages you want to send: '))
    print('\nEncryption:\n')
    ciphertexts = []
    for _ in range(num_messages):
        plaintext = input('Enter plaintext: ').lower().replace(" ", "")
        ciphertext = cipher.encrypt(plaintext, key)
        ciphertexts.append(ciphertext)
        print('Cipher Text:', ciphertext)

if __name__ == "__main__":
    main()
```

- ✓ **Output :**



```
 >_ pwsh  6  1ms
● ›› python -u "c:\Users\Tushar\Documents\SEM 6\INS\CODES\6\6_2.py"
  Vigenere cipher
  Generated Key: fzxrex

  Enter the number of messages you want to send: 1

  Encryption:

  Enter plaintext: tushar
  Cipher Text: ytpyeo
```