

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 6

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.

Code :

```
import random as r
import YSL_io as ysl
ysl.printMGNTA('\n\tOne-time Pad Method\n\t')

def enc(pt, key):
    global a
    ct = ''
    kl = len(key)
    ptl = len(pt)
    if kl < ptl:
        key += pt[ptl-kl:]
        kl = ptl
    for i in range(ptl):
        ct += chr(((a[pt[i]] + a[key[i]]) % 26) + 97)
    ysl.printRED(f'Cipher text of given plaintext ', end='')
    print(pt, end='')
    ysl.printRED(' : ', end='')
    print(ct, end='\n\n')
    return ct, key

def dec(ct, key):
    global a
    dt = ''
    kl = len(key)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 6

```
    ct1 = len(ct)
    if k1 < ct1:
        key += ct[:ct1-k1:]
        k1 = ct1
    for i in range(ct1):
        dt += chr(((a[ct[i]] - a[key[i]]) % 26) + 97)
    ysl.printRED(f'Decipher text of given ciphertext ', end='')
    print(ct, end='')
    ysl.printRED(' : ', end='')
    print(dt, end='\n\n')

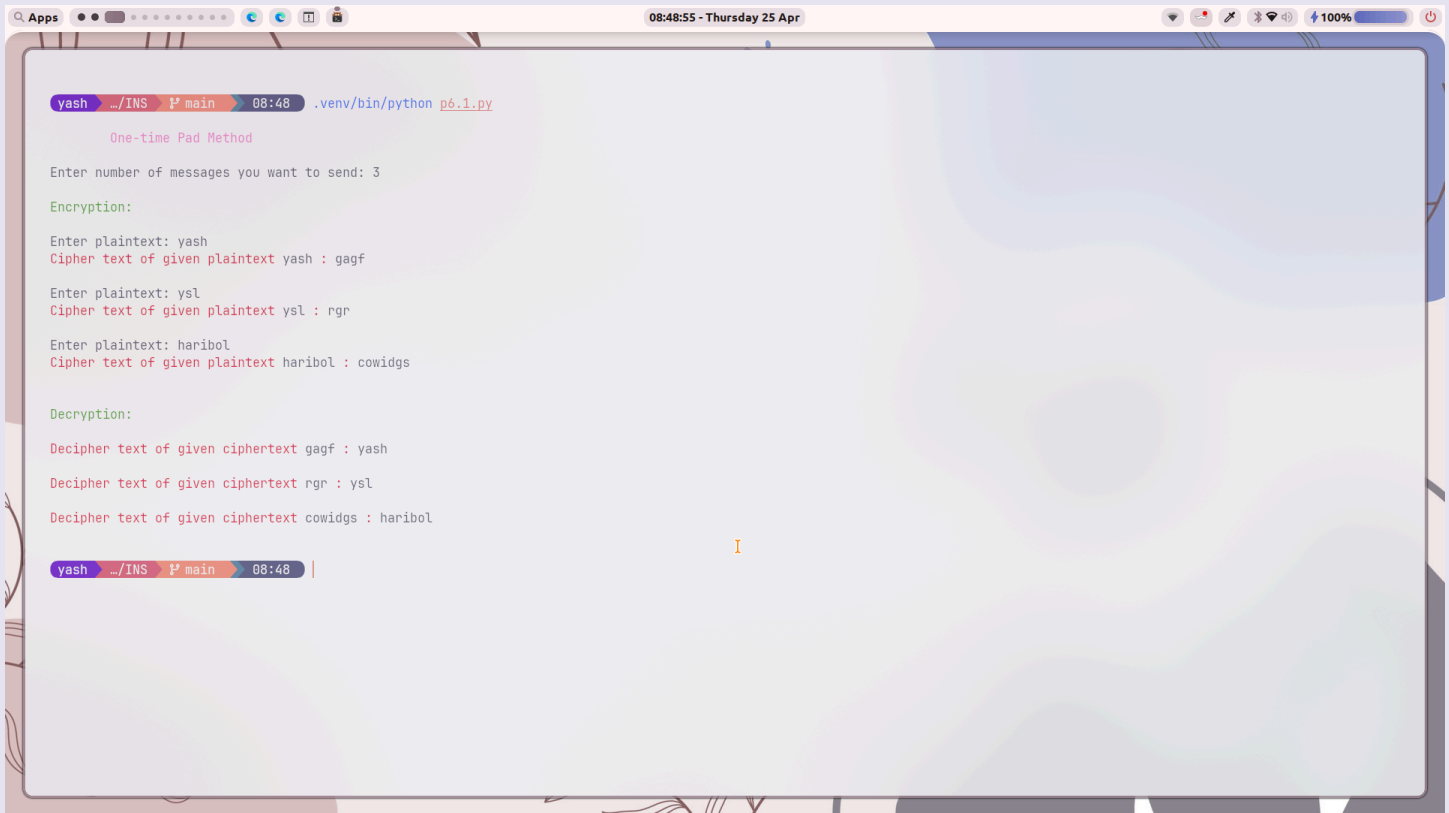
a = {chr(x+97):x for x in range(26)}
n = int(input('Enter number of messages you want to send: '))
keys = []
plaintext = []
ciphertext = []

ysl.printGRN('\nEncryption:\n')
for _ in range(n):
    pt = input('Enter plaintext: ').lower().replace(" ", "")
    plaintext.append(pt)
    key = ''
    for x in range(r.randint(6, 10)):
        key += chr(97 + r.randint(0, 25))
    ct_key = enc(pt, key)
    ciphertext.append(ct_key[0])
    keys.append(ct_key[1])

ysl.printGRN('\nDecryption:\n')
for i in range(n):
    dec(ciphertext[i], keys[i])
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 6

Output :



```
yash ~/INS main 08:48 .venv/bin/python p6.1.py

One-time Pad Method

Enter number of messages you want to send: 3

Encryption:

Enter plaintext: yash
Cipher text of given plaintext yash : gagf

Enter plaintext: ysl
Cipher text of given plaintext ysl : rgr

Enter plaintext: haribol
Cipher text of given plaintext haribol : cowidgs

Decryption:

Decipher text of given ciphertext gagf : yash

Decipher text of given ciphertext rgr : ysl

Decipher text of given ciphertext cowidgs : haribol

yash ~/INS main 08:48 |
```

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

Code :

```
import YSL_io as ysl
k_lst = []

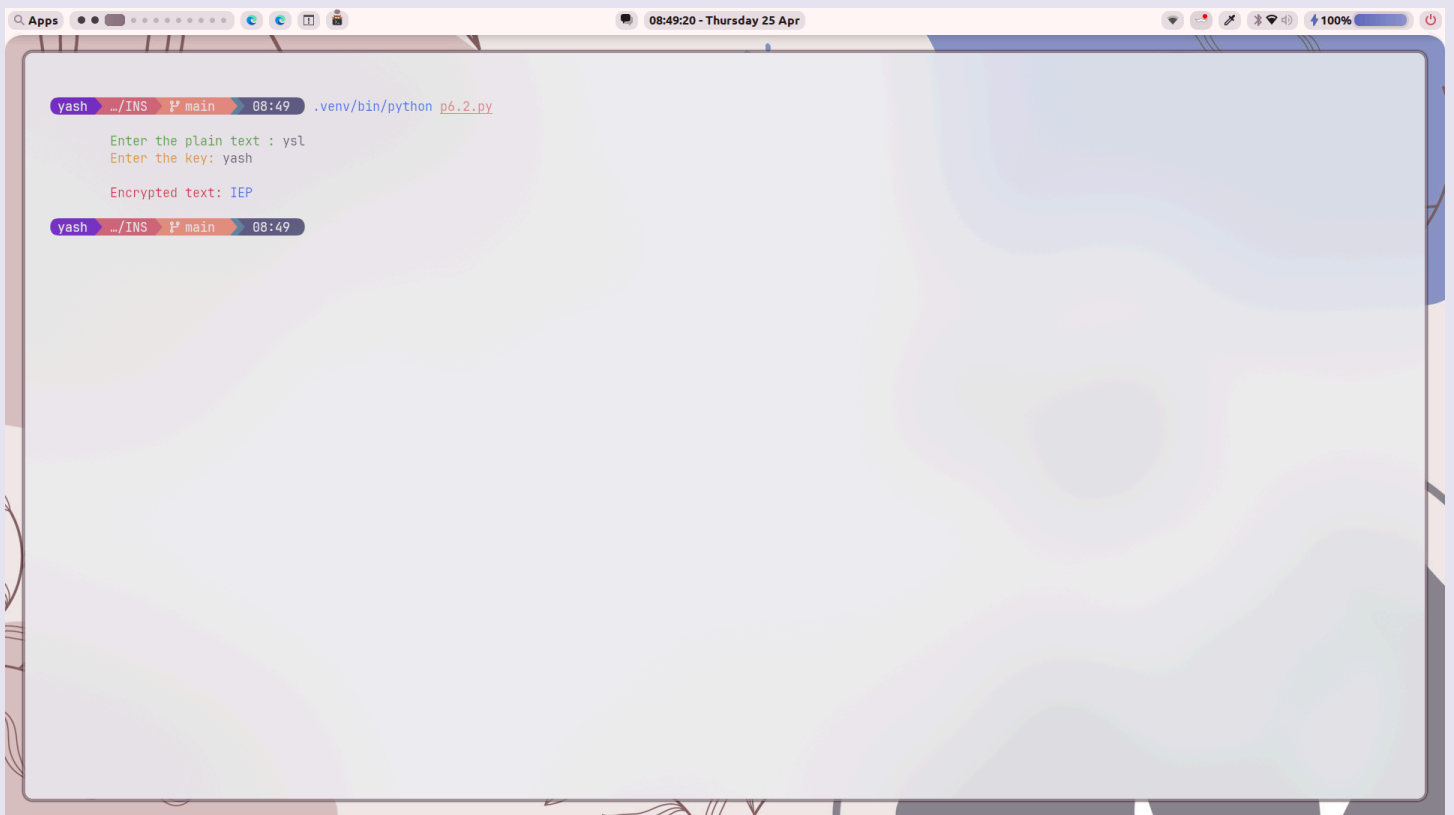
def enc_vig(plain, k):
    enc_txt = ''
    for i in range(len(plain)):
        char = plain[i]
        k_char = k[i]
        enc_char = chr((ord(char) + ord(k_char)) % 26 + ord('A'))
        enc_txt += enc_char
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
INS Practical 6

```
    return enc_txt

pt = ysl.inputGRN("\n\tEnter the plain text : ")
k = ysl.inputORNG("\tEnter the key: ")
ysl.printRED(f'\n\tEncrypted text: ', end='')
ysl.printBLU(enc_vig(pt, k))
```

Output :



The screenshot shows a terminal window with a macOS-style title bar. The terminal prompt is 'yash ~/INS main 08:49'. The user has run the command '.venv/bin/python p6.2.py'. The program's output is as follows: 'Enter the plain text : ysl' (in green), 'Enter the key: yash' (in orange), and 'Encrypted text: IEP' (in red). The terminal window also shows system status icons at the top right, including signal strength, battery level at 100%, and a power button icon.

```
yash ~/INS main 08:49 .venv/bin/python p6.2.py
Enter the plain text : ysl
Enter the key: yash
Encrypted text: IEP
yash ~/INS main 08:49
```