```
================================================================================
        Name        :Lalita Sharma
        RNo         :06


================================================================================
                        Experiment :01
                        ***************
Aim:Write a program to implement Transposition and One time Pad Cipher.


Program:

# ****************************************************************************
def Encrypt(mainlist,row,col):
    k=0
    cipherlist=[]
    list1=[]
    for z in range(int(row)):
        if k<int(col):
            for list in mainlist:
                list1.append(list[k])
            cipherlist.append(list1)
            list1=[]
            k+=1
    return cipherlist

# ****************************************************************************


def Decrypt(clist,row,col):
    k=0
    ret=[]
    list2=[]
    for z in range(int(row)):
        if k<int(col):
            for list in clist:
                list2.append(list[k])
            ret.append(list2)
            list2=[]
            k+=1
    return ret

# ****************************************************************************
def listtostr(list):
    listtostr=''
    for li in list:
        listtostr+=''.join(li)
    return listtostr
# ****************************************************************************
def bitlisttp(string):
    list1=[]
    for char in string:
        ascii=ord(char)
        if ascii>=65 and ascii<=90:
            alphindx=ascii-65
        else:
            alphindx=ascii-97
        list1.append(int(alphindx))
    return list1
```

```python
# *************************************************************************
def Addbitlisttpfun(plainbitlist,keybitlist):
    ret=[]
    for i in range(len(plainbitlist)):
        ret.append(plainbitlist[i]+keybitlist[i])
    return ret
# *************************************************************************
def cipherlisttpfun(addedbitstp):
    ret=[]
    for num in addedbitstp:
        if (abs(num)+97)>122:
            ret.append(chr(97))
        else:
            ret.append(chr(abs(num)+97))
    return ret


# *************************************************************************
def decryptedlisttpfun(addedbitstp,keybitlisttp):
    subtractedbits=[]
    # print("addedkey list:",addedbitstp)
    # print("keybit lsit:",keybitlisttp)
    for i in range(len(addedbitstp)):
        subtractedbits.append(addedbitstp[i]-keybitlisttp[i])
    finallist=[]
    # print("subtracted bit list:",subtractedbits)
    for num in subtractedbits:
        # if chr(num+97)
        finallist.append(chr(num+97))
    return finallist
# *************************************************************************
def Transposition():
    mainlist=[]
    li=[]
    i=0
    j=0
    row,col=input("enter row and column(with space in between)\n").split(',')
    string=input("Enter the message\n")
    # print(len(string))
    total=int(row)*int(col)
    if len(string)!=total:
        string=string + ' '*(total-len(string))
    # print(len(string))
    for char in string:
        if i<int(row):
            li.append(char)
            if(len(li)==int(col)):
                mainlist.append(li)
                i+=1
                li=[]
    # print(mainlist)
    cipherlist=Encrypt(mainlist,row,col)
    ciphertext=listtostr(cipherlist)
    print("Encrypted string:  ",ciphertext)
    decryptedstr=listtostr(Decrypt(cipherlist,row,col))
    print("Decrypted string:   ",decryptedstr)


# *************************************************************************
def TimePad():
    string=input("Enter the message\n")
```

```
        print("Key must be less than or equal to plaintext in length.")
        print("Plaintext must not contain numbers.\n")
        key=input("enter the key\n")
        i=0
        if len(key)<len(string):
            nooftimes=len(string)-len(key)
            while i< nooftimes:
                for char in key:
                    if len(key)==len(string):
                        break
                    key=key+char
                i+=1
        # print(key)
        plainbitlisttp=bitlisttp(string)
        # print(plainbitlisttp)
        keybitlisttp=bitlisttp(key)
        # print(keybitlisttp)
        addedbitstp=Addbitlisttpfun(plainbitlisttp,keybitlisttp)
        # print(addedbitstp)
        cipherlisttp=cipherlisttpfun(addedbitstp)
        ciphertexttp=listtostr(cipherlisttp)
        print("Encrypted text: ",listtostr(ciphertexttp))
        decryptedlisttp=decryptedlisttpfun(addedbitstp,keybitlisttp)
        print("Decrypted text: ",listtostr(decryptedlisttp))

# ****************************************************************************


def main():

    while(1):
        inp=int(input("\nChoices:\n1: Transposition\n2: TimePad\n3: Quit\n"))
        if inp==1:
            Transposition()
        elif inp==2:
            TimePad()
        else:
            print("Terminating.......")
            break
# ****************************************************************************
main()


************************* END *********************************************

   OUTPUT
   ==========
```

```
PS C:\Users\91705> & python c:/Users/91705/Documents/5_sem/CN/EXP01.py

Choices:
1: Transposition
2: TimePad
3: Quit
1
enter row and column(with space in between)
4,4
Enter the message
hello how are u?
Encrypted string:   howee   lhaulor?
Decrypted string:    hello how are u?

Choices:
1: Transposition
2: TimePad
3: Quit
2
Enter the message
how is everything going on?
Key must be less than or equal to plaintext in length.
Plaintext must not contain numbers.

enter the key
abc
Encrypted text:  hpyajuafxesatiknhagpknhaooa
Decrypted text:  how is everything going on?

Choices:
1: Transposition
2: TimePad
3: Quit
3
Terminating.......
PS C:\Users\91705>
```