
Name : Atharva Paliwal

Roll No : 40 [5B]

*** EXPERIMENT NO: 01 ***

AIM- Write a Programme to implement transposition and one time pad cipher

CODE-

TRANSPOSITION CIPHER

#Atharva Paliwal

```
def transposition(msg,rows,cols):          #Encryption and Decryption Function
    k=0                                     #index of text message
    arr = [[0]*cols for j in range(rows)]  #matrix initialisation
    for i in range(rows):
        for j in range(cols):
            if k>=len(msg):                #if matrix size exceeds length of message
                arr[i][j]=' '              #giving spaces
                continue
            arr[i][j]=msg[k]
            k=k+1
    print('Matrix :')                      #printing matrix
    for i in range(rows):
        print(*arr[i])
```

```

newmsg=''
for i in range(cols):          #adding message column-wise
    for j in range(rows):
        newmsg=newmsg+arr[j][i]
return newmsg

msg=input('Enter Your Message : ')
rows, cols =map(int,input('Enter size of row and column : ').split())
encrmsg=transposition (msg,rows,cols)
print('Encrepted Message:',encrmsg)
decrprmsg=transposition (encrmsg,cols,rows)
print('Decrepted Message:',decrprmsg)

*****

```

OUTPUT-

```

Enter Your Message : Hello Mr How are You
Enter size of row and column : 5 4
Matrix :
H e l l
o   M r
   H o w
   a r e
   Y o u
Encrepted Message: Ho   e HaYlMorolrweu
Matrix :
H o
e   H a Y
l M o r o
l r w e u
Decrepted Message: Hello Mr How are You

```

ONE-TIME PAD CIPHER

#function for finding index of required text

```
def find(bit_map,k):  
    for i in range(len(bit_map)):  
        if k==bit_map[i]:  
            return i
```

#function for encryption

```
def encrpt(msg,key):  
    #Initialising a list of alphabets,numbers and space  
    bit_map=list(chr(i)for i in range(65,91))+list(chr(i)for i in  
    range(97,123))+list(str(i)for i in range(0,10))+[' ']  
    #increasing key length if key length is smaller  
    key=key*(len(msg)//len(key)+1)  
    encr_msg=''  
    for i in range(len(msg)):  
        #modulo addition of message with key  
        encr_msg=encr_msg+  
        bit_map[(find(bit_map,msg[i])+find(bit_map,key[i]))%63]  
    return encr_msg
```

#function for decryption

```
def decrpt(msg,key):  
    bit_map=list(chr(i)for i in range(65,91))+list(chr(i)for i in  
    range(97,123))+list(str(i)for i in range(0,10))+[' ']
```

```

key=key*(len(msg)//len(key)+1)

decr_msg=''

for i in range(len(msg)):

    #subtraction of encrypted msg and key

    k=find(bit_map,msg[i])-find(bit_map,key[i])

    k=k+[0,63][k<0]  #if difference is negative adding 63

    decr_msg=decr_msg+bit_map[k]

return decr_msg

```

#Driver Code

```

msg=input('Enter Your Message : ')
key=input('Enter the Key : ')
encr_msg=encript(msg,key)
print('Cipher Text : ',encr_msg)
decr_msg=decrpt(encr_msg,key)
print('Decrpt Text : ',decr_msg)

```

OUTPUT-

```

Enter Your Message : Hello Mr How Are You
Enter the Key : Apple7
Cipher Text :  HIPLH6MVosHs pVEdUoY
Decrpt Text :  Hello Mr How Are You

```
