Assignment No.1

Code:-

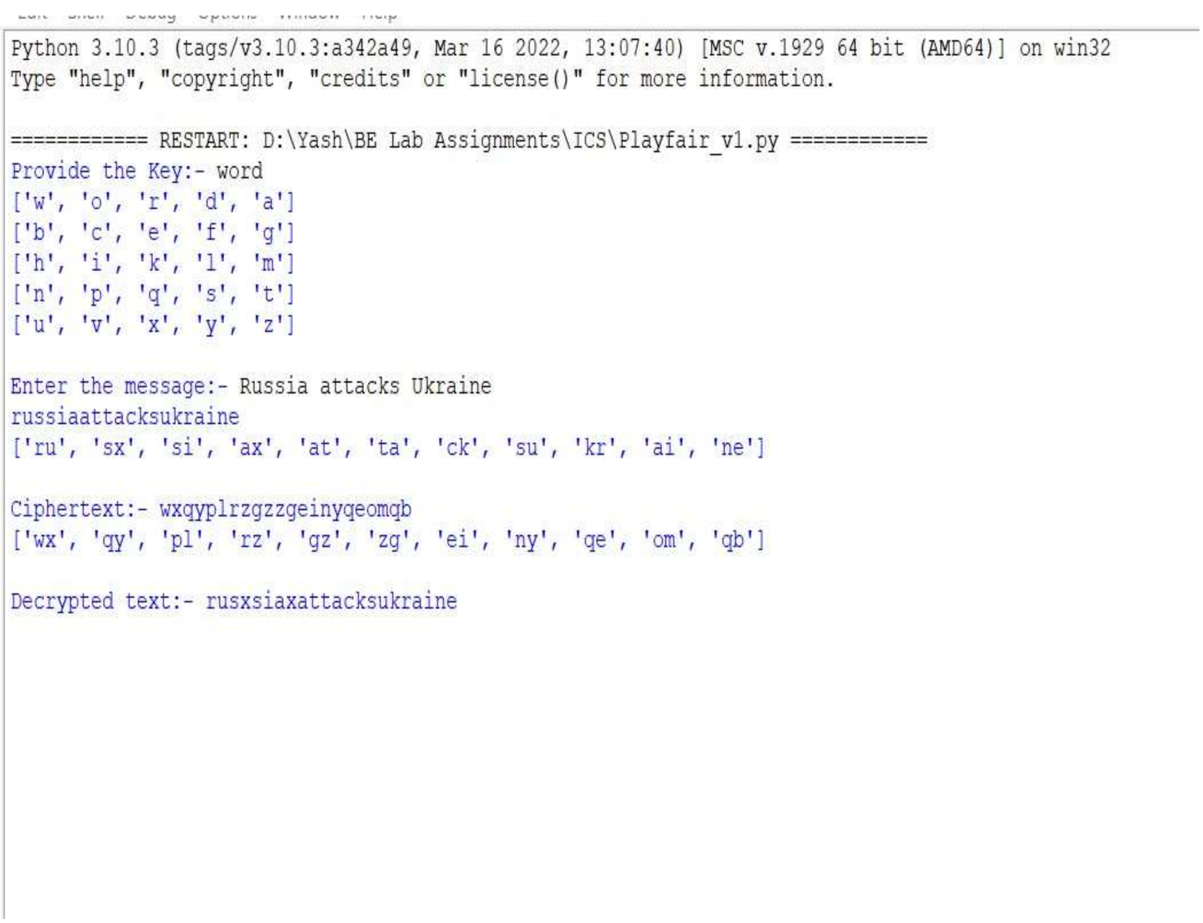
```
key=input("Provide the Key:- ")
key=key.replace(" ","").lower()
kset=list()
for i in key:
    if i not in kset:
        kset.append(i)
l1=list()
l2=list()
for i in kset:
    l2.append(i)
    if(len(l2)==5):
        l1.append(l2)
        l2=list()
c=97
while(c<=122):
    if(chr(c) not in kset and c!=106):
        l2.append(chr(c))
        if(len(l2)==5):
            l1.append(l2)
            l2=list()
    c+=1
for i in l1:
    print(i)
message=input("\nEnter the message:- ")
message=message.replace(" ","").lower()
print(message)
def encryption(msg):
    pairs=list()
    l=len(msg)
```

```
i=0
while(i<len(msg)):
    if(msg[i]!=msg[i+1]):
        pairs.append(msg[i:i+2])
    else:
        msg=msg[0:i+1]+'x'+msg[i+1:]
        pairs.append(msg[i:i+2])
    i+=2
    if(i==len(msg)-1):
        msg=msg+'x'
print(pairs)
ans=list()
for x in pairs:
    dict1=dict()
    dict1={'x1':0,'y1':0,'x2':0,'y2':0}
    for i in range(5):
        for j in range(5):
            if(x[0]==l1[i][j]):
                dict1['x1']=i
                dict1['y1']=j
            if(x[1]==l1[i][j]):
                dict1['x2']=i
                dict1['y2']=j
    #print(l1[dict1['x1']][dict1['y2']]+l1[dict1['x2']][dict1['y1']])
    if(dict1['x1']==dict1['x2']):
        ans.append(l1[dict1['x1']][(dict1['y1']%4)+1]+l1[dict1['x1']][(dict1['y2']%4)+1])
    elif(dict1['y1']==dict1['y2']):
        ans.append(l1[(dict1['x1']%4)+1][dict1['y1']]+l1[(dict1['x2']%4)+1][dict1['y2']])
    else:
        ans.append(l1[dict1['x1']][dict1['y2']]+l1[dict1['x2']][dict1['y1']])
return "".join(ans)
```

```
def decryption(msg):
    pairs=list()
    l=len(msg)
    i=0
    while(i<len(msg)):
        if(msg[i]!=msg[i+1]):
            pairs.append(msg[i:i+2])
        else:
            msg=msg[0:i+1]+'x'+msg[i+1:]
            pairs.append(msg[i:i+2])
        i+=2
        if(i==len(msg)-1):
            msg=msg+'x'
    print(pairs)
    ans=list()
    for x in pairs:
        dict1=dict()
        dict1={'x1':0,'y1':0,'x2':0,'y2':0}
        for i in range(5):
            for j in range(5):
                if(x[0]==l1[i][j]):
                    dict1['x1']=i
                    dict1['y1']=j
                if(x[1]==l1[i][j]):
                    dict1['x2']=i
                    dict1['y2']=j
        #print(l1[dict1['x1']][dict1['y2']]+l1[dict1['x2']][dict1['y1']])
        if(dict1['x1']==dict1['x2']):
            if(dict1['x1']==4 or dict1['x2']==4):
                ans.append(l1[dict1['x1']][(dict1['y1']%5)-1]+l1[dict1['x1']][(dict1['y2']%5)-1])
            else:
```

```
        ans.append(l1[dict1['x1']][(dict1['y1']%4)-1]+l1[dict1['x1']][(dict1['y2']%4)-1])
    elif(dict1['y1']==dict1['y2']):
      if(dict1['y1']==4 or dict1['y2']==4):
        ans.append(l1[(dict1['x1']%5)-1][dict1['y1']]+l1[(dict1['x2']%5)-1][dict1['y2']])
      else:
        ans.append(l1[(dict1['x1']%4)-1][dict1['y1']]+l1[(dict1['x2']%4)-1][dict1['y2']])
    else:
      ans.append(l1[dict1['x1']][dict1['y2']]+l1[dict1['x2']][dict1['y1']])
  return "".join(ans)
cipher_txt=encryption(message)
print('\nCiphertext:-',cipher_txt)
decrypted=decryption(cipher_txt)
print('\nDecrypted text:-',decrypted)
```

Output:-



```
Python 3.10.3 (tags/v3.10.3:a342a49, Mar 16 2022, 13:07:40) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

============ RESTART: D:\Yash\BE Lab Assignments\ICS\Playfair_v1.py ============
Provide the Key:- word
['w', 'o', 'r', 'd', 'a']
['b', 'c', 'e', 'f', 'g']
['h', 'i', 'k', 'l', 'm']
['n', 'p', 'q', 's', 't']
['u', 'v', 'x', 'y', 'z']

Enter the message:- Russia attacks Ukraine
russiaattacksukraine
['ru', 'sx', 'si', 'ax', 'at', 'ta', 'ck', 'su', 'kr', 'ai', 'ne']

Ciphertext:- wxqyplrzgzzgeinyqeomqb
['wx', 'qy', 'pl', 'rz', 'gz', 'zg', 'ei', 'ny', 'qe', 'om', 'qb']

Decrypted text:- rusxsiaxattacksukraine
```

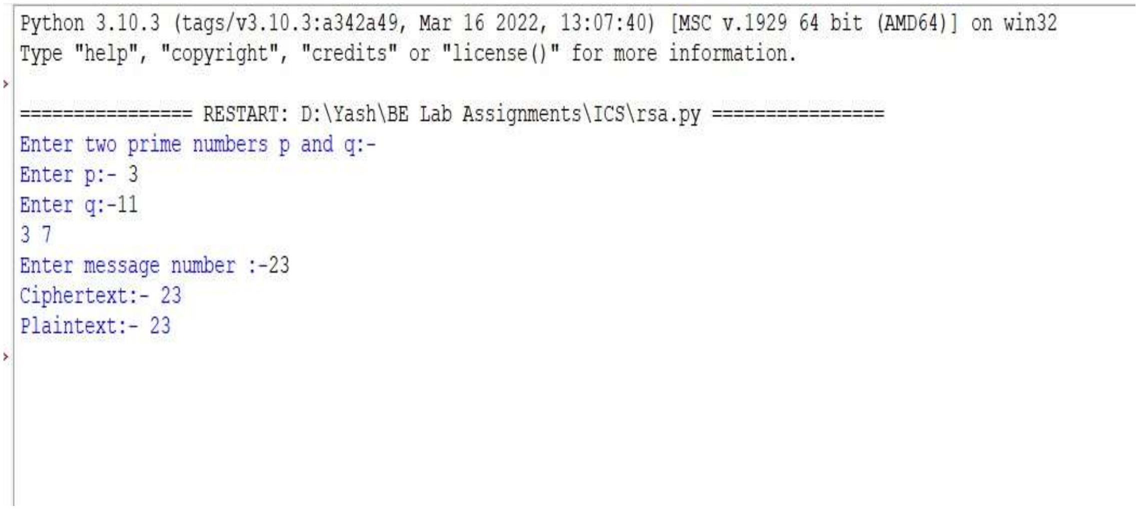
Assignment No 2

Code:-

```
print('Enter two prime numbers p and q:-')
p=int(input('Enter p:-'))
q=int(input('Enter q:-'))
n=p*q
theta=(p-1)*(q-1)
def gcd(x,y):
   if(x<y):small=x
   else:small=y
   for i in range(1,small+1):
      if((x%i==0) and (y%i==0)):gcd=i
   return gcd
e=0
for i in range(2,theta):
   if(gcd(i,theta)==1):
      e=i
      break
x=1
d=0
while(1):
   if((theta*x+1)%e==0):
      d=int((theta*x+1)/e)
      break
   x+=1
pb_key=(e,n)
pv_key=(d,n)
print(e,d)
msg=int(input('Enter message number :-'))
cipher=(pow(msg,pb_key[0])%pb_key[1])
print('Ciphertext:-',cipher)
```

```
decrypt=(pow(cipher,pv_key[0])%pv_key[1])

print('Plaintext:-',decrypt)
```

Output:-



```
Python 3.10.3 (tags/v3.10.3:a342a49, Mar 16 2022, 13:07:40) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

================ RESTART: D:\Yash\BE Lab Assignments\ICS\rsa.py ================
Enter two prime numbers p and q:-
Enter p:- 3
Enter q:-11
3 7
Enter message number :-23
Ciphertext:- 23
Plaintext:- 23
```

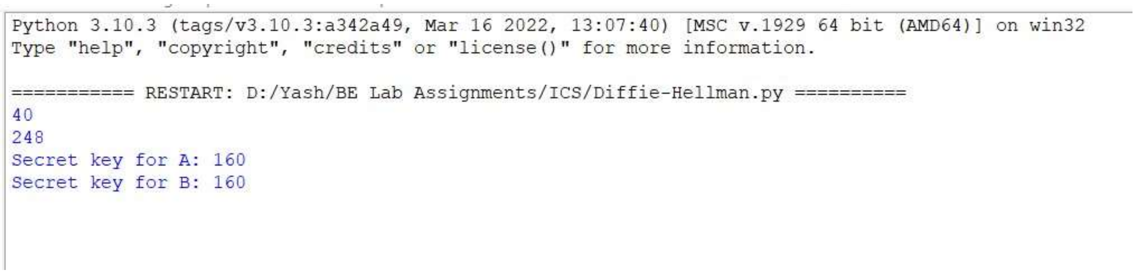
Assignment 3

Code:-

```
# Public Values (Everyone can see this)
q = 353
alpha = 3

def calculate_y_value(X: int) -> int:
    Y = (alpha ** X) % q
    return Y

# User A
Xa = 97  # user selection
Ya = calculate_y_value(Xa)
print(Ya)

# User A sends values (q = 353, alpha = 3, Ya = 40) to User B
# Anyone can know these values

# User B
Xb = 233
Yb = calculate_y_value(Xb)
print(Yb)

# Send public value back (Yb = 248)

def generate_key(X: int, Y: int) -> int:
    K = (Y ** X) % q
    return K

# User A
Ka = generate_key(Xa, Yb)
```

```
print("Secret key for A:",Ka)

# User B
Kb = generate_key(Xb, Ya)
print("Secret key for B:",Kb)

assert Ka == Kb
```

Output:-



```
Python 3.10.3 (tags/v3.10.3:a342a49, Mar 16 2022, 13:07:40) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

=========== RESTART: D:/Yash/BE Lab Assignments/ICS/Diffie-Hellman.py ==========
40
248
Secret key for A: 160
Secret key for B: 160
```