Name: Shubham Jaisali
Course: BCA "B" 6th Sem
Roll no. 57
Subject Information Security and Cyber laws.

Ques3> Vignere Cipher

```
def generateKey(String, key):
    Key = list(Key)
    if len(string) == len(Key):
        return(Key)
    else:
        for i in range(len(string) - len(Key)):
            Key.append(key[i % len(key)])
    return("".join(key))

def encryption(string, key):
    encrypt_text = []
    for i in range(len(string)):
        x= (ord(string[i]) + ord(Key[i])) % 26
        x+= ord('A')
        encrypt_text.append(chr(x))
    return("".join(encrypt_text))

def decryption(encrypt_text, key):
    orig_text=[]
    for i in range(len(encrypt_text)):
        x =(ord(encrypt_text[i] - ord(Key[i] + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return("".join(orig_text))
```

Shubham

```
if __name__ == "__main__":
    string= "Cryptography"
    Keyword= "Monarchy"
    Key= generateKey(string, Keyword)
    encrypt_text = encryption(String, Key)
    print("Encrypted text: ", encrypt_text)
    print("Decrypted text: ", decryption(encrypt_text,Key))
```

Shubham

Name: Shubham Jaisali

Rollno. 57 (1121139)

Course BCA "B" 57 6th sem

Subject Information Security and Cyber laws.

Ques4) Program to Generate OTP

```
import math, random
def generateOTP()
        digits = "0123456789"
         OTP = " "
        for i in range (4):
                OTP += digits[math.floor(random.
                                    random() * 10)]

         return OTP
if __name__ == "__main__" :
        print("OTP of 4digits: ", generateOTPC))
```

Shubham

Name: Shubham Jaisali

Course BCA 6th sem "B"

Roll no. 57 (1121139)

Subject Information Security and Cyber laws.

Ques 5. Encryption and decryption using Caesar Cipher

```
def encrypt(text, s);
        result=" "
        for ~~f~~ i in range(len(text)):
            char=text[i]
            if(char.isupper()):
                result+= chr((ord(char)+ s-65)%26
                                            +65)
            else:
                result+= chr((ord(char)+s-97)%26
                                         +97)

        return result

def decrypt(text, s):
        result=" "
        for i in range(len(text)):
            char=text[i]
            if(char.isupper()):
                result= result + chr((ord(char)
                                  -s-65)%26+65)

            else~~f~~:
            result += ~~r~~ chr((ord(char)-s-97)
                                  %26+97)

        return result
```

L

```
text= "Attack from the North"
S=3
print("Text : " +text)
print("shift : " + str(s))
encrypt_text = encrypt(text, s)
print("****** Encryption ****** ")
print("Plain text : " +text)
print("Cipher text : " + encryp_text)
print("******Decryption***** ")
print("Ciphertext : " + encrypt_text)
print("Plain text : " +, decrypt(encrypt_text, s))
```