

NAME- YOGENDRA SINGH ASWAL

UNIVERSITY ROLL NO - 1121174

SUBJECT - INFORMATION SECURITY & CYBER LAWS LAB

SUBJECT CODE - PBC 601

4.

```
import math, random
```

```
def generateOTP():
```

```
    string = '0123456789abcdefghijklmnopqrstuvwxyz  
             yzABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

```
    OTP = ""
```

```
    length = len(string)
```

```
    for i in range(6):
```

```
        OTP += string[math.floor(random.random() *  
                                length)]
```

```
    return OTP
```

```
if __name__ == "__main__":
```

```
    print("OTP of length 6:", generateOTP())
```

5.

* Encryption using Ceaser cipher

```
def encrypt(string):
```

```
    cipher = ''
```

```
    for char in string:
```

```
        if char == ' ':
```

```
            cipher = cipher + char
```

```
        elif char.isupper():
```

```
            cipher = cipher + chr((ord(char) + 3 - 65) %  
                                   26 + 65)
```

```
        else
```

```
            cipher = cipher + chr((ord(char) + 3 - 97) %  
                                   26 + 97)
```

```
    return cipher
```

```
text = text "Attack from North"
```

```
print("Original string:", text)
```

```
print("after encryption:", encrypt(text))
```

* Decryption using Ceaser Cipher

```
def decrypt(string):
```

```
    plain = ''
```

```
    for char in string:
```

```
if char == ' ':
```

```
    plain = plain + char
```

```
elif char.isupper():
```

```
    plain = plain + chr((ord(char) - 3 - 65) % 26 + 65)
```

```
else:
```

```
    plain = plain + chr((ord(char) - 3 - 97) % 26 + 97)
```

```
return plain
```

```
text = 'Dwrwdfn iurp @wv'
```

```
text = 'Dwrwdfn rq @ruwk'
```

```
print("cipher string", text)
```

```
print("after decryption:", decrypt(text))
```

3.

```
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)
```

```
    cipher_text = []
```

```
    for i in range(len(string)):
```

```
        x = (ord(string[i]) + ord(key[i])) % 26
```

```
        x += ord('A')
```

```
        cipher_text.append(chr(x))
```

```
    return "".join(cipher_text)
```

```
def decryption(cipher_text, key):
```

```
    org_text = []
```

```
    for i in range(len(cipher_text)):
```

```
        x = (ord(cipher_text[i]) - ord(key[i]) + 26) % 26
```

```
        x += ord('A')
```

```
        org_text.append(chr(x))
```

```
    return "".join(org_text)
```

```
if __name__ == "__main__":
```

```
    string = "cryptography"
```

```
    key = "Monarchy"
```

```
keyw = generateKey (String, key)
```

```
ciphertext = encryption (String, keyw)
```

```
print ("ciphertext:", ciphertext)
```

```
print ("original / Decrypt text:" decryption (ciphertext  
    , key)
```