Name - Ashish Dabral

Course - BCA '6th' 'A'

University Roll No - 1171027

Subject :- Information ~~Cyber~~ Security and cyber laws

Ashish

Scanned by TapScanner

Q3 Write a program for the encryption and decryption of the Vigenere Cipher on the plain text = "Cryptography" with a key = "Monarch"

Sol (3)

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

Encryption

```
def cipherText(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))
```

Abhisth

Scanned by TapScanner

Decryption

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

Driven Code

```
if __name__ == "__main__":
    String = "Cryptography"
    keyword = "Monarchy"
    Key = generateKey(string, keyword)
    cipher_text = cipherText(string, Key)
    print("Ciphertext:", cipher_text)
    print("original/Decryted Text:", originalText(cipher_text
    , Key))
```

Ashish

Scanned by TapScanner

Q4 Write a program to implement OTP(One time password)

Sol

(4)

```
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 4 digits:", generateOTP())
```

Ashish

Scanned by TapScanner

Q5 Write a program to implement encryption and decryption using Caesar Cipher on the input plaintext = "ATTO Attack from North"

Sol

(5)

```
def encrypt(text, s):
    result = ""

for 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
text = "Attack from North"
s = 3
print("text:"+text)
print("shift:"+str(s))
print("Cipher:" + encrypt(text,s))
```

Ashish

Scanned by TapScanner

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

text = "Attack from North"
    s = 3
print("cipher:" + decrypt(text,s))
print("shift:" + str(s))
print("text:" + text)
```

Ashish

Scanned by TapScanner