INFORMATION SECURITY AND CYBER LAWS

NAME – SALONI MAGGO

COURSE – BCA 6 B

ROLL NO. – 39

UNIVERSITY ROLL NO. – 1121120

Exam – Practical Exam (End Sem)

Email – salonimaggo2001@gmail.com.

Father Name – Mr. Kuldeep Kumar Maggo

Saloni

Scanned by CamScanner

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

def decryption ( encrypt_text , Key ) :
    orignal_text = []
    for i in range ( len ( encrypt_text )) :
        x = (ord ( encrypt_text [i]) - ord ( Key [i] )) % 26
```

Saloni

Scanned by CamScanner

```
    x = x + ord ('A')
    orignal_text . append ( chr(x))
  return ("  " . join ( orignal_text ))

if  __name__ == "__main__":
   s = " Cryptography "
   string = s. upper ()
   Keyword = " Monarchy "
    Key = generatiKey ( string , Keyword )
    encrypt_text = encryption ( string , Key )
    print (" Encrypted text is : " , encrypt_text )
    print ( " Orignal / Decrypted Text : ", decryption ( encrypt_text ,
                                                        Key )) .
```

Output -

```
Encrypted text is : OLRVQW TVMJAE
Orignal / Decrypted Text : CRYPTOGRAPHY .
```

Saloni

Q4.

```
import math , random
 def funcOTP () ;
  x = "0123456789"
  OTP = " "
  for i in range (16):
  OTP = OTP + x[math. floor ( random. random () * 10 )]
  return OTP
  if __name__ == "__main__":
   print (" OTP of 16 digits : " , funcOTP ())
```

OUTPUT

```
OTP of 16 digits : 5239577340121692
```

Saloni

Scanned by CamScanner

Q5.

```
def encrypt ( string , shift ) :
  cipher = "
  for char in string :
    if char == ' ' :
      cipher = cipher + char
    elif char . isupper ( ) :
       cipher = cipher + chr ( ( ord ( char ) + shift - 65 )
                 % 26 + 65 )
    else :
       cipher = cipher + chr ( ( ord ( char ) + shift - 97 )
                 % 26 + 97 )
  return cipher
  text = " Attack from North "
  s = 3
  print ( " orignal string : ", text )
  print ( " after encryption : ", encrypt ( text , s ) )

def decrypt ( string , shift ) :
  plain = "
  for char in string :
```

Scanned by CamScanner

```
if char == ' ':
  plain = plain + char
elif char.isupper():
  plain = plain + chr((ord(char) - shift - 65)
                % 26 + 65)
else:
  plain = plain + chr((ord(char) - shift - 97)
                % 26 + 97)
return plain

text = encrypt(text, s)
s = 3
print(" after decryption : ", decrypt(text, s))
```

output —

```
  Original string :   Attack from North
  after encryption:   Dwwdfn  iurp  Qruwk
  after decryption:   Attack  from  North.
```

Saloni

Scanned by CamScanner