VIKAS KUMAR
BCA "C" 6 SEM
1121165

Ans I

```
(3) def generate Key (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  ciphertext (string ,key)
       cipher_text = [ ]
       for i in range ( len (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))
```

Vikas

```
def orignal . Text ( 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 ))
 if  __name__ == " __main__ ":
    string = ( cryptography "
   key word = " monarchy" ;
 - key = generatekey ( string , key word )
  cipher _ text = cipher text ( string , key)
 print ( "cipher text : " , cipher _ text )
  print ( orignal / Decrypted Text :" )
  Original Text ( cipher _ text , key )
```

Ans (5)

```
def encrypt (text ,s)
  result = " "
  for i in range (len (text)):
  char = text [i]
  if (char . isupper [ ]):
  result += chr ((ord (char)+ s-97 )% 26+97)
  return result
  text = " Attack from North
   s= 4
print " shift key : " + str (s)
print " Cipher : " + encrypt (text ,s)
```

Ans (4) **WAP to implement OTP**

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

if ___name___ == "__main___":
print ("OTP of 4digits:" generator OTP ())
```

Vikey.