Name - Abhishek Badoni
Course - BCA 'A' sem 6
University roll no - 1121002

(Information security & cyber laws
TBC 601)

MCQ

Ans 1 => Asymmetric key encription with sender public key

Ans 2 => Spyware

Ans 3 => An authentication of an electronic record

Ans 4 => CyberLaws

Ans 5 => Only an alphanumeric

Ans 6 => Idea is same title is different

Ans 7 => checksum

Ans 8 => The identity of the character is changed while its position remains unchanged.

Ans 9 => both b and c

Ans 10 => none

Name Abhishek Badoni (Information Security & Cyber laws)
Course - BCA 'A' sem 6 TBC 601
University roll no - 1121002 Date 15/6/2021

Q.5 Ans:) # Implementation of encryption & Decryption using Caesar cipher

```
=> def encryption (Plain_text, key):
       encryption = ""
   for c in Plain text:
       if c.isupper():
       c_index = ord(c) ord('A')
       c_shifted = (c_index+key) %26+ord('A')
       c_new = chr(c_shifted)
       encrypted += c_new

   def c.islower():
       c_index = ord(c)-ord('a')
       c_shifted = (c_index+key)%26
       +ord['a']
       c_new chr(c_shifted)
       encrypted += c_new
```

```
dif c.isdigit():
    c_new = (int(c)+key)%10
    encrypted+=str(c_new)

else: encrypted+=c

return encrypted

def decryption(ciphertext,key):
    decrpted = ""
    for c in ciphertext:
        if c.isupper():
            c_index=ord(c)-ord('A')
            c_og_pos=(c_index-key)%26+ord('A')
            c_og=chr(c_og_pos)
            decrypted+=c_og
        dif c.islower():
            c_index=ord(c)-ord('a')
            c_og_pos=(c_index-key)%26+ord('a')
            c_og=chr(c_og_pos)
            decrypted+=c_og
```

```
dif c.is digit():
        c_og = (int(c) - key) % 10
        decrypted += str(c_og)
    else:
        decrypted += c
    return decrypted

Plain_text = "Attack from north"
Ciphertext = encryption(Plaintext, 4)
Print("Plain text message: \n" Plain_text)
Point("Encrypted Ciphertext: \n", ciphertext)
    decryptedmsg = decryption(ciphertext, 4)
Point("The decrpted message is: \n", decrptedmsg)
```

Name - Abhishek Badoni (Information security & cyber laws TBC 601)
Course - BCA 'A' sem 6
University roll no - 1121002 Date 15/6/2021

Q. 3 Ans =>

```
# Vigenere 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))
# encryption
def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) + ord(key[i]))
        x += ord('A')
```

```
        ciber_text append (chr(x))
    return ("join(ciber_text))
# function for decrypting
def original text (cipher_text, key):
    orgi orgi_text = []
    for i in range (an(cipher_text)):
        x = (ord (cipher_text[i])-ord(key[i])+26)
                                        %26
        x+ = ord ('A')
        orig text append chr (x))
    return ("" orig_text))
# Driver Code
if __name__ == "__main__":
    String = "cyptography"
    keyword = "mantchy"
    key = genetatekey (Sting keyword)
Print ("ciber text:" cipher_text)
Print (original/Decrypted text:",
        original text (cipher_text_key)
```

Name- Abhishek Badami (Information security & Cyberlaws)
Course- BCA 'A' Sem 6 (TBC 601)
University-rollno- 1121002
Student ID - 18211216 Date- 15/06/2021

Q.4 Ans ⇒

```
# import ~~library~~ random as r
# function for OTP generation
def OTPgen():
    OTP = ""
    for i in range(4):
        OTP += str(r.randint(1,9))
Print("Your one time Password is")
Print(OTP)
OTPgen()
```