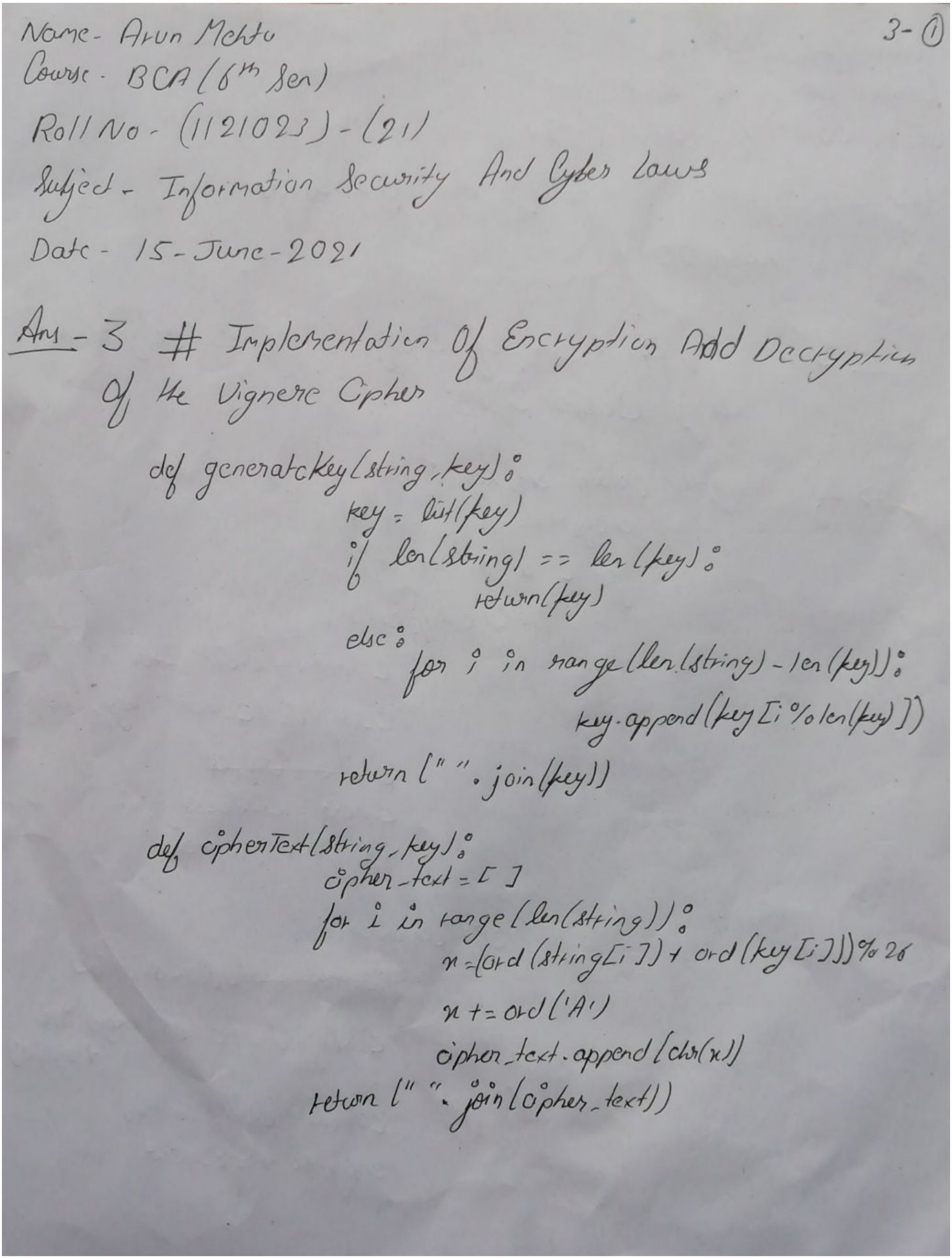
Name- Arun Mehta
Course- BCA (6th Sem)
Roll No- (1121023) - 21
Subject- Information Security And Cyberlaws
Date : 15 - June - 2021

MCQs

1. Asymmetric key encryption with sendes public key
2. Spyware
3. An authentication of an electronic record
4. cyber laws
5. Only an alphanumeric
6. Idea is same title is different
7. Checksum
8. The identity of the character is changed while its position remains unchanged.
9. Both b and c
10. None



Name- Arun Mehta <span>3-①</span>

Course - BCA (6th Sem)

Roll No - (1121023) - (21)

Subject - Information Security And Cyber Laws

Date - 15-June-2021

Ans - 3 # Implementation Of Encryption Add Decryption Of the Vignere 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))

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))
```

```
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))


if __name__ == "__main__":

        string : " "
        keyword : "
        key = generatedKey(string, keyword)
        cipher_text = cipherText(string, key)
        print("Ciphertext : ", cipher_text)
        print("Original/Decrypted Text : ",
                        originalText(cipher_text, key)
```

Name- Arun Mehta
Course- BCA (6th Sem)
Roll No.-(1121023)-(21)
Subject- Information Security And Cyber Laws
Date- 15- June- 2021

Ans- 4. # Implement Of OTP

```
import random as r
def ~~opt~~ otpgenerate():
        otp=" "  # Empty String
        for i in range(4):
                otp+= str(r.randint(1, 9))
        print("Your OTP is :")
        print(otp)

~~otpgen()~~
otpgenerate()
```

Name- Arun Mehta
Course - BCA (6th Sem)
Roll NO- (1121093) -(21)
Subject = Information security And CyberLaws
Date - 15- June - 2021

Ans - 5 # Implementation Of Encryption And Decryption Using Caeser Cipher

```
=> def encryption(plain_text, key):
        encrypted= " "
        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~~
                c_new = chr(c_shifted)
                encrypted+ = c_new

            elif c.islower():
                c_index= ord(c)-ord('a')
                c_shifted=(c_index +key)%26
                                   + ord('a')

                c_new = chr(c_shifted)
                encrypted + = c_new
```

5-ⓐ

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

        else:
            encrypted += C


    return encrypted


def decryption( ciphertext, key):
            decrypted= " ".
            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
                    elif 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
        elif c.isdigit():
                c_og = (int(c) - key) % 10
                decrypted += str(c_og)

        else:
              decrypted += c

    return decrypted

plain_text = "Attack from North"
ciphertext = encryption(plain_text, 4)
print(" Plain text message :\n", plain_text)
print("Encrypted Ciphertext :\n", ciphertext)
decryptedmsg = decryption(ciphertext, 4)
print(" The decrypted message is : \n", decryptedmsg)
```