

HBHAs BHAT
BCA 'A'

1121001

Cyber Security Lab

MCQ

Q1

Ans1 Asymmetric Key encryption
with Sender Public Key

Q2

Ans2 Spyware

Q3

Ans3 An Authentication of an
electronic record

Q4

Ans4 Cyberlaws

Q5

Ans5 ~~idea is~~ only on Alphamumeric
~~Some title is different~~

Q6

Ans6 idea is same but title is different

Q7

Ans7 has no value

Q8

Ans8 The identity of character
is changes which its position
remain unchanged.

Q9

Ans9 both b & c

Q10

Ans10 none

ABHAS BHATT
BCA 'A'

Abhas

1121001

Cyber Security

Q1

Ans 1] Do a Security checkup

Go to Security checkup to get personalized Security Recommendation for your google account including.

• 1.1] Turn on 2-Step Verification

1.2] Turn on Screen lock.

1.3] Update account recovery options

2] Update your Software.

If your browser, operating system, or apps are out of date the software might not be safe from hackers.

2.1] Update your browser, apps & Operating system.

3] Use unique, strong password.

It's risky to use the same passwords on multiple sites. If Password for one site is leaked it could be used to get into your accounts for multiple sites.

3.1] Manage your password

3.2] Protect your password from hacker.

4] Protect against Suspicious Messages. & content hackers can be via email, text message, phone call & web page to pretend to be ^{institution, family etc} ~~institutions~~

4.1] Avoid suspicious request

4.2] Avoid suspicious web page

4.2] Avoid suspicious email

ABHAS BHATT

BCA 'A'

1121001

Cyber Security

OTP Program

Q4

Ans4 #import library

import math.random

function to generate OTP

def generate OTP():

Declare a digital variable

which stores all digits

digits = "0123456789"

OTP = ""

for i in range(n):

OTP += digits [math.floor (random.random () * 10)]

return OTP

Driver code

if __name__ == "__main__":

Print ("OTP of 4 digits:", generate OTP (1))

ABHAS BHATT
BCA '6'

Abhas

1121001

Cyber Security

Q 5

Sol 5 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_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

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')~~

NBHAS BHATT

1121001

BCA 'A'

Cyber law

Abhas

Q5

Ans 5

```
def decryption (ciphertext, Key):  
    decrypted = ""  
    for C in ciphertext:  
        if C.isupper():  
            C_index = ord(C) - ord('A')  
            C_orig_pos = (C_index - Key) % 26 + ord('A')  
            C_orig = chr(C_orig_pos)  
            decrypted += C_orig  
        elif C.islower():  
            C_index = ord(C) - ord('a')  
            C_orig_pos = (C_index - Key) % 26 + ord('a')  
            C_orig = chr(C_orig_pos)  
            decrypted += C_orig  
        elif C.isdigit():  
            C_orig = (int(C) - Key) % 10  
            decrypted += str(C_orig)  
        else:  
            decrypted += C  
    return decrypted  
  
Plain_text = "Attack From North"  
Ciphertext = encryption(Plain_text, 4)  
print("Plain text message", Plain_text)  
print("Encrypted ciphertext is:", Ciphertext)  
decryptedmsg = decryption(Ciphertext, 4)  
print("The decrypted message is:", decryptedmsg)
```