Name- Himanshu Gusain
Course:- BCA (SC)
Roll no:- 1121061 (05)

Que 1:- (b) Asymetric key encryption with user public key.

Que 2:- (a) Software

Que 3:- (c) An authentication of an electronic record.

Que 4:- (b) Cyber security

Que 5:- (b) only on ASCII coded data

Que 6:- (c) All

Que 7:- (a) hash value

Que 8:- (b) The identity of the character is changed while its position remain unchanged

Que 9:- (b) To make even no. of letters

Que 10:- (a) Total length of word.

- Descriptive:-

Que 1:- (1) Go to security checkup to get personalized security recommendations for your google account, including:-

(a) Add or update account recovery options:- Your recovery phone number and email address are powerful security tools.

(b) Alert if there's suspicious activity on your account.

(c) Recovery your account if you are ever locked.

(2) Step verification helps prevent a hacker from getting into your account if your account, even if they steal your password. To avoid common phising techniques associated with text messages codes, choose a stronger second verification step:-

(a) Update your browser.

(b) learn how to update your chromebook's O.S.

(c) learn how to update your android version.

(3) Use unique strong password. It's risky to use the same password on multiple sites. If your password for one site is hacked, it could be used to get data your accounts for multiple sites.

Que 4:- Python :-

```
4 digit Numeric OTP
# import library
  import math, random
# function to generate OTP
    def generateOTP():
# Declare a digits variable
# which stores all digits
    digits = "0123456789"
    OTP = ""
" length of password can be changed"
" by changing value in range"
    for i in range(4):
    OTP += digits[math.floor(random.
                random() * 10)]
    return OTP
  # Driver code
  if __name__ == "__main__":
print(" OTP of 4 Digits :", generate
                OTP())
```

Output

OTP of 4 Digit : 3211

Ques:1 using C:2

```
#include <stdio.h>
#include <string.h>
int main()
{
char msg[]="THECRAZYPROGRAMMER";
char key[]="HELLO"
int msglen=strlen(msg), keylen=strlen(key), i, j;
char newkey[msglen], encryptedmsg[msglen], decryptedmsg[msglen];
for(i=0, j=0; i<msglen; ++i, ++j){
    if(j==keylen)
        j=0;
    newkey[i]=key[j];
}
newkey[i]='\0';
//encryption
for(i=0; i<msglen; ++i)
encryptedmsg[i]=((msg[i]+newkey[i])%26)+'A';
encryptedmsg[i]='\0';
//decryption
for(i=0; i<msglen; ++i)
decryptedmsg[i]=(((encryptedmsg[i]-newkey[i]+26)%26)+'A');
decryptedmsg[i]='\0';
printf("original message: %s", msg);
printf("\nKey: %s", key);
printf("\nNew Generated key: %s", newkey);
printf("\nEncrypted message: %s", encryptedmsg);
printf("\nDecrypted Message: %s", decryptedmsg);
return 0;
}
```