Name - Jasmeet Kaur Course - BCA 6 'C' Roll no - 1121067

Ans. MCQs : (10)

1) Asymmetric key encryption with Sender public key

2) Spyware.

3) An authentication of on electronic record.

4) Cyber Security

5) Only on ASCII Coded data

6) All

7) hash value

8) The identify of character is changed white its position.

9) to make even no. of better.

10) total length of word.

Name:- Jasmeet kaur   Course- BCA 6 'c'

Rollno = 10   University Rollno - 1121007

Paper Code - PBC 601   Subject -

Ans-1. Three Security aspects of the Google Account:

Step1:- Add or Update Account Recovery options.

Your recovery phone number & email address are powerful security tools. This Contact info Con be used to help:

- Block Someone from using your account without your permission

- Alert your if there's suspicious activity on your account.

- Recovery your accant if your account if you're ever locked out.

☆ 2 - Step Verification

- Security keys (Most Secure verification step.)

- Google prompts (More secure than text message Codes)

Step2:- Update your Softwares

If your browser, operating sly or apps are

out of-date, the software might not be safe from hackers.

o Update your browser

→ Make sure you're using latest version.

☞ Update your Operating sly.

o Update your chrome books.

Step 3: - Use Unique strong Password.

· It is risky to use the same password or multiple sites. If your password for one site is hacked, it could be used to get into your accounts for multiple sites.

Name - Jasmeet Kaur
Rollno - 10
Paper Code → PBC 601
Course - BCA 6'c'
University Rollno - 1121067
Subject - Information Security

Ans 4)
```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main()
{ char plantxt[100], otp[100];
printf(" enter plan text\n");
fflush(stdin);
fgets(plantxt), size of(plantxt), stdin);
print f("enter otp txt of length %d\n", strlen(plantxt));
fflush(stdin);
fgets(plantxt, sizeof(plantxt), stdin);
printf("enter otp txt of length %\n", strlen(plantxt));
fflush(stdin);
fgets(otp, sizeof(otp), stdin);
for(int i=0; i<strlen(plantxt); i++)
{ if(isupper(plantxt[i]))
{
otp[i] = toupper(otp[i]);
if(plantxt[i] + (otp[i] - 'A' >'z')){
plantxt[i] = planttxt[i] + (otp[i] - 'A') - 26};
```

```c
if (plantxt[i] + (otp[i] - 'A') <= 'z')
    {
    plantxt[i] = plantxt[i] + (otp[i] - 'A');
    }

}
else if (islower (plantxt[i]))
    {
    otp[i] = tolower (otp[i]);
    if (plantx[i] + (otp[i] - 'a') > 'z')
    {
    plantx[i] = plantx[i] + (otp[i] - 'a') -
26;
    }
    if (plantxt[i] + (otp[i] - 'a') <= 'z')
    {
    plantxt[i] = plantxt[i] + (otp[i] - 'a');
    }
    else {
    plant xt[i] = plant xt[i]; }
    }
print f (" Cypher txt \t %-s \n", plantxt);
return 0;
}
```

5.

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char message = "ATTACK FROM NORTH",ch,
        messaged);
    int i, key;
    printf(" Enter key");
    scanf("%d", &key);
    for(i=0; message[i] != '\0';i++)
    {
        ch = message[i];
        if(ch>='a' && ch<='z')
        {
            ch = ch+key;
            if(ch> 'z')
            {
                ch= ch-'z' + 'a'-1;
            }
            message[i]=ch;
        else if(ch>='A' && ch<='z')
        {
            ch= ch + key;
```

```c
        if (ch>'z')
    {
        ch=ch-'z'+'A'-1;
    }
        message [i] =ch';
}
}
    printf ("encrypted message is %\n', message);
    for(i=0; message [i] !='\o', i++)
    {
        ch d2 messaged [i];
        if (ch>='a' && ch<z'z')
        {
            ch=ch- key;
            if (ch<'a)
            {ch =ch+'z' - 'a' +1;
            }
            messaged [i] =ch;
        }
        else if (ch>'A' && ch<'z)
        {ch = ch - key;
            if (ch<'A')
            {ch =ch+z' - 'A' +1;
            }
            messaged [i] =ch;
        }
    print f ("decrypted message is %s', messaged)
        return 0;
    }
```