```
Lab-01
 AIM: - Write a programme for shift cipher
          and monoalphabetic Substitute cipher
          with craptanalysis.
(a) Shift Cipher
# molude < bits/stdc#+, h>
using numespace std;
string encoupt (string plaintext, int k);
string decrypt (string cipherText, int k);
void cryptunalysis (string cipherText);
string encrypt (string plaintext, int K)
     string cipherText = " ! ! lemety string
      int array
      for (m+ i=0; i < plainText : length(); ++i)
            cipherText[i] = PlainText[i] - a.
          cipherText [i] = (cipherText[i]+k)-1.26;
cipherText += (cipherText[i] + 'q');
     cout << "cither text: "<< cirherText << end ;
```

```
return cipherText;
String decrypt (string either Text, int k).
     PlainTeatti] = cipherTeatti] - a;
     PlainText[i] = PlainText[i] - k;
     it ( PlamText[i] <0)
          Plumfest [i]+=26;
     PlanText[i] 1.=26;
     PlainText += (PlainText[1]+ 'a');
     return plaintext; "<< Plaintext << end!
void constanalysis (string ciphertext).
      string PlainText = 1111.
      too (mt k=0 j k < 26; k++)
           for (int i=0 / < (irherText length(); +xi)
              PlainText[i]= ciPhorText[i]-'q1'
              Plaintext[i] = plantoxt[i]-k;
              if (PlainText[i] <
```

```
void cryptanalysis (string cirhertext)
      saring plaintent = "1";
      for (m+ k=0; K<26; K++)
            PlainText = decrypt (cipherText, K):
            cout << 'key: "<< k << "pecroppled Text!"
                          << plainText << entli
int main ()
    String Plain Text;
    cin77 PlainText >7 K;
    cout << "PlainText: " << plainText << end;
    string circherteset = energet (plainteset, K)
    PlainText = decrypt (cipher Text);
  return o;
  input
  Key = 3 , text = munut
 output:
 cipher Text : PX9di
 Decrypted text; muncif.
```

6 (6) monoalphabetic Substitute Cipher code :-#include < bitk/stdc++.h7 using numespace std; String encrypt (string plain Text, smingk); string decrypt (string ciphertext, string k); string enought (string PlainText, murchan, chan key) strong apportant = 1111. too (int 1=0; (PlainText. length o; ti) cipherText += key[pluinText[1]]; cout << "cipher Text: " << cirherText << end! return cipherTest: void decrypt (string cipherText, maps char, char) key) string PlainText = 1111. for (m+ 1=0; i < cipher Text , length c); ++i) Plaintext + = key[ciphorText[i]]

cout << 'Demosted text: "<<plainText<< end!; int main() 11 this is a random key map < char, char > kest string plainText, cm>> PlainTent; cout << "original mag: " << plaintent < cend!; string cipherText = encrypt (PlainText, key); decry et (cipherText, key); return o' I net: muneit output: Original text: mynest Cipher text: nfmzu Decrypted text: munuf.