```
Lab-03
Aim: Write a program to implement
 1. Autokey Cipher
 2. Playfuir Cipher
Program: -1: Autokey Cipher
Source code:-
#Include < bits (stoc++ih)
using namespace sta;
string Generate key (int k, string plain Text)
    string key = "";
    key += (char) ('a'+ K1);
    for ( int i = 0; i < plainte et geneth (); ++i)
        Key += PlainTex+[i];
    return key;
String Energetion ( string koy, string plainText)
 string cipherText="11.
  for (int i=0; ix PlainText. longth (); ++i)
     cipherText += ((PlainText [i]-'a1) +
  (key[i]-'a')) .1.26;
ciPherText[i] = ciPherText[i] + 'q';
```

```
string Decryption (string key, string cipherTent)
    String plaintent = " ";
   for (int i=0; i< cipherTealt. length(); ++i)
      PlainTest += ((cirherTent[i]-'al)-
                     (Key[i]-41));
      if ( Plantad(i) <0)
         PlainTex+(i]+=26;
 Plantent[]:1=20;
Plantent[]= plaintent[]+ al;
   return plantent;
int main ()
  int KI.
  string PlainTeret
  cm>> plainText>> KI;
 string key = Generatekey (KI, PlainText);
  string cipherText = Eneryption ( key, plainText);
 cout << "encrypted-Text: " << ciphertext << end!
· string decrypted Text = Demption ( key, ciphertext)
 cout << "decrypted Text; "<< decrypted Text < cond!
 return o.
```

	input output
	munestdivan Encrypted text: - Puhnfildun
	3 pergeted text: munuflian
	Activities Principle v Land of Maria
	Program-2:- Playfair Cirher
	Tarlahas basis For Taladaris I.
	source code:
	+ mclude < bits / stdc++.h>
	using namespace sto.
	Holder N 5
	Void Generate key (string KI, char key[H][N])
	Pair (int)
	Pairkchur, bool > alphubets (26);
- 1	forcint i=0; i<26; ++i)
	1 1 1 1 5 5 5 6 1 1 1 1 1 1
	alphabets[i] = make-pair('a'+i, true);
	m+ K=0, x=0;
	for (inti=0; i <n; i+i)<="" td=""></n;>
	4
	foo(int j=0; j <n; +ti)<="" td=""></n;>
	7
	if (KKK1. length())
	1 VICVT - VICVT ·
	Key[i][j] = KI[K]

```
int m = 1 ([x] - 'al,
   It ( KICK) == " | | KICK] == ")
      alphabets[q]. second = false;
      al Phabets [8]. second = fulse;
   alphabets[m]. seemd = fulse;
   K++
else
   it ( 8 L 26 gr al phubet [ 8] second)
    Key[i][i] = alphabets[x]. first;
      alphabets [8]. second = false;
       it(8==8)
         alphubets [++>]. second = felse;
```

```
pair < 1 nt > securch ( char ch, cherr key[M][N])
      it(ch== 1)
      ch='i';
    Pair</nt> index;
    for (m+ 1=0; i(N; ++1)
      tox( mt ) = 0; J<N; ++1)
          it ( key[i][s] == ch)
                  index = make - pair (1, i);
    return Moder.
String Encryption (charkey[N][N], string plain Text)
    string circhertext = "";

pair< int, int> Indoxi, mdex2;

for(int i = 0; i< plaintext.longht(); ++)
        mdex1 = search(plamTex+[i), key);
        moderal = search ( PlamText [i+1], key);
       it index! tisst = = indexe2. first)
           moderch. second = (indeay. second +1).1. st;
           index 2. second = (index2. second +1) .1. N.
```

```
cipherToxt += key[indexcl. first] [indexl. seomd];
        CIPHONTEN + = Key[mden2.first] [inden2.secons];
     else it (index). second == index2 second)
         modera, trost = (index1, first+1) 1. N;
         index 2. +18st = (index 2, fix + 1) .1. N.
        ciPhenText += 1xpy[, mdex1, frit][Index1, Second];
        a phentered to key [ indend, fost] [indener. socond]
    else
         cipherText += key[index]. first][mdex]. second].
         cipherText += key[mdex2.first][index2. second]
 return cipherText;
Strong Decorption (char Key[N] [N], string cipherioa)
    String plainText ="1";
    purk int inty modes 1, index 2;
    for (int i=0; i < ciPherText. length(); i+=2)
          index 1 = secret (cipherText(i), key).
          Index 2 = seemsh (righer Text (1), Key);
```

```
if ( index 1, first = = index2.first) 1
      index1: second --:
     indexez second -- ',
     If ( mdex). second < 0)
        index1. second += st.
     if (indexes second(0)
        index2. second +7N's
     indescy, sociond 1-= N;
     index2. second 1 = N';
     PlainToxt += Key[indoxi, first][indexi, second];
     PlainTend += key [index2: first] [mdex2. second];
 else it (index). samd == index2. sacond)
    index list = -;
      index2. first --;
      it (index 1. first <0)
      index! trest += N'
     it (mdex2, frit<0)
     index2. first += N',
     index 1. first 1. = N;
     mdex 2. tryt 1. = N.
    PlainText + = key[mdex1. Hist][mdexel. second]
    PlainText += Key [ indexe fixt] [indexs. second]
else
  Plaintext += Key[mdex1. Fix+][mdex2. Second]:
  elamient + = key [ index2. fist] [index2 sound].
```

