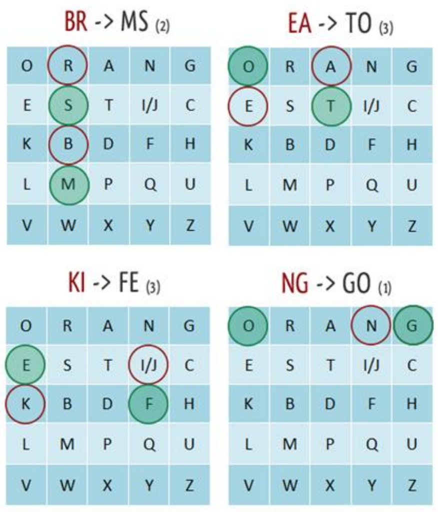
Program – 3

AIM: To implement a program to show encryption and decryption through Play-Fair Cipher.

Introduction and Theory

Playfair cipher, also known as Playfair square, Wheatstone-Playfair cipher or Wheatstone cipher is a manual symmetric encryption technique and was the first literal digram substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In this scheme, pairs of letters are encrypted, instead of single letters as in the case of simple substitution cipher.

In playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I. The sender and the receiver decide on a particular key, say 'Algorithm'. In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order.



Algorithm

- First, a plaintext message is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter. Let us say we want to encrypt the message "Programming". It will be written as – Pr og ra mm in gZ
- The rules of encryption are -
  - If both the letters are in the same column, take the letter below each one (going back to the top if at the bottom)
  - If both letters are in the same row, take the letter to the right of each one (going back to the left if at the farthest right)
  - If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Program – 3

Code

```
#include <bits/stdc++.h>
using namespace std;
class Playfair
{
	string key;
	int find_in_key(char a){
		for(int i=0 ; i<25 ; i++){
			if(a == key[i])
				return i;
		}
		return -1;
	}
	pair<char, char> encipher_pair(char a, char b){
		int a_id = find_in_key(a), b_id = find_in_key(b);
		int x_a=a_id/5, y_a=a_id%5, x_b=b_id/5, y_b=b_id%5;

		if(x_a == x_b)
			return make_pair(key[x_a*5 + (y_a+1)%5], key[x_b*5 + (y_b+1)%5]);
		if(y_a == y_b)
			return make_pair(key[((x_a+1)%5)*5 + y_a], key[((x_b+1)%5)*5 + y_b]);
		else
			return make_pair(key[x_a*5 + y_b], key[x_b*5 + y_a]);
	}
	pair<char, char> decipher_pair(char a, char b){
		int a_id = find_in_key(a), b_id = find_in_key(b);
		int x_a=a_id/5, y_a=a_id%5, x_b=b_id/5, y_b=b_id%5;

		if(x_a == x_b)
			return make_pair(key[x_a*5 + (y_a-1)%5], key[x_b*5 + (y_b-1)%5]);
		if(y_a == y_b)
			return make_pair(key[((x_a-1)%5)*5 + y_a], key[((x_b-1)%5)*5 + y_b]);
		else
			return make_pair(key[x_a*5 + y_b], key[x_b*5 + y_a]);
	}
public:
	Playfair(string KEY){key = KEY;}

	string encipher(string plainText)
	{
		if(plainText.length()%2 == 1)
			plainText += "x";
		string cipherText = plainText;
		for(int i=0 ; i<plainText.length() ; i+=2){
			pair<char, char> t = encipher_pair(plainText[i], plainText[i+1]);
			cipherText[i] = t.first;
			cipherText[i+1] = t.second;
		}
```

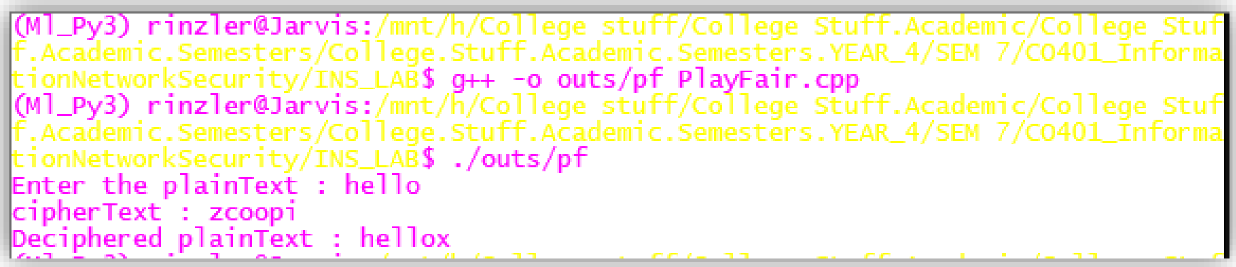
Program – 3

```cpp
                return cipherText;
        }
        string decipher(string cipherText)
        {
                if(cipherText.length()%2 == 1)
                        cipherText += "x";
                string plainText = cipherText;
                for(int i=0 ; i<cipherText.length() ; i+=2){
                        pair<char, char> t =
decipher_pair(cipherText[i], cipherText[i+1]);
                        plainText[i] = t.first;
                        plainText[i+1] = t.second;
                }
                return plainText;
        }
};

int main()
{
        string key = "alonpzmihxvyrswukdfteqgcb";
        Playfair p(key);
        string plainText;
        cout << "Enter the plainText : ";
        cin >> plainText;
        cout << "cipherText : " << p.encipher(plainText) << endl;
        cout << "Deciphered plainText : " <<
p.decipher(p.encipher(plainText)) << endl;
}

```

Results and Outputs:



```
(Ml_Py3) rinzler@Jarvis:/mnt/h/College stuff/College Stuff.Academic/College Stuf
f.Academic.Semesters/College.Stuff.Academic.Semesters.YEAR_4/SEM 7/CO401_Informa
tionNetworkSecurity/INS_LAB$ g++ -o outs/pf PlayFair.cpp
(Ml_Py3) rinzler@Jarvis:/mnt/h/College stuff/College Stuff.Academic/College Stuf
f.Academic.Semesters/College.Stuff.Academic.Semesters.YEAR_4/SEM 7/CO401_Informa
tionNetworkSecurity/INS_LAB$ ./outs/pf
Enter the plainText : hello
cipherText : zcoopi
Deciphered plainText : hellox
```

Findings and Learnings:

1. We have implemented Playfair ciphers