Practice 3: CLASSICAL SYMMETRIC ENCRYPTION

3.1 OVERVIEW

3.1.1 Introduction

- Lab 3: PlayFail Cipher and Transposition Cipher
- Practice time: class: 3 study hours, self-study: 3 study hours.

3.1.2 Objective

- This course provides students with knowledge of cryptographic algorithms and how they are used in today's world.
- The content emphasizes the principles, topics, approaches, and problem solving related to the underlying technologies and architectures of the field.

3.2 CONTENTS

3.2.1 Basic knowledge

The Playfair cipher, the first literal digram substitution cipher, is a manual symmetric encryption method. Although Charles Wheatstone came up with the idea in 1854, Lord Playfair is credited with popularizing it.

The Playfail algorithm is to create a 5x5 matrix based on a given keyword and letters of the alphabet:

- First write the words of the keyword in the rows of the matrix starting from the first row.
- If the matrix is empty, write other letters of the alphabet that have not been used in the remaining cells. It is possible to write in a pre-conventional sequence, such as from the beginning of the alphabet to the end.
- Since there are 26 English letters, one box is missing. Usually we put two words in a common box, for example I and J.

Assume to use keyword MONARCHY. Construct the corresponding Playfair key matrix as follows:

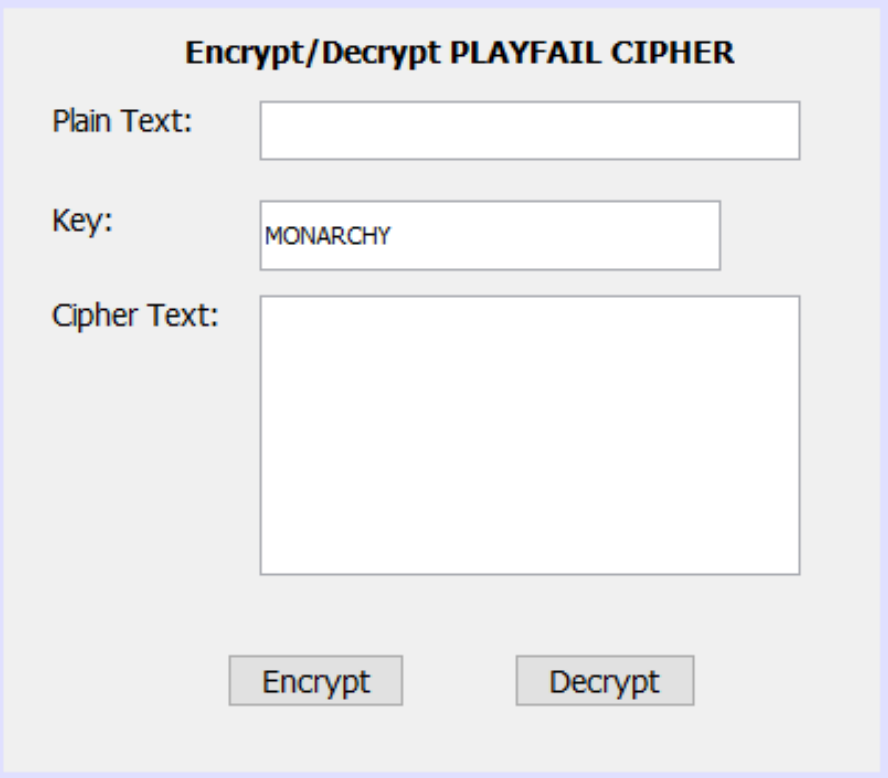
M	O	N	A	R
C	H	Y	B	D
E	F	G	I,J	K
L	P	Q	S	T
U	V	W	X	Z

3.2.2 Playfail Algorithm

Write a program to encrypt and decrypt text with Playfail encryption algorithm. The program can perform the following functions:

- Allow text input into the system.
- Allows entering text protection keys.

❖ **Step 1: Design Form:**



❖ **Step 2: Write code for initialization function:**

```
public class frmPLAYFAIR extends javax.swing.JFrame {
    char pf[][]={{'M','O','N','A','R'},
                 {'C','H','Y','B','D'},
                 {'E','F','G','I','K'},
                 {'L','P','Q','S','T'},
                 {'U','V','W','X','Z'}
                };

    public frmPLAYFAIR() {
        initComponents();
        this.txtKey.disable();
    }
```

❖ **Step 3: Write an event handler function:**

- **3.1 Write code for functions:**

```
private String Encrypt(String banro){
    int n=banro.length();
    int i=0;
    String banma="";
    char a,b;
    while (i<n) {
        if (i==n-1) {
            a=banro.charAt(i);
            b='X';
            i++;
        }else{
            a=banro.charAt(i);
            b=banro.charAt(i+1);
            if (a==b) {
                b='X';
                i++;
            }else
                i+=2;
        }
        banma+=Replace(a,b);
    }
    return banma;
}
```

```
String Replace(char a,char b)
{
    String vta=FindLocation(a);
    String vtb=FindLocation(b);
    char x,y;
    if (vta.charAt(0)==vtb.charAt(0)) {
        x=pf[vta.charAt(0)-'0'][((vta.charAt(1)-'0')+1)%5];
        y=pf[(vtb.charAt(0)-'0')][((vtb.charAt(1)-'0')+1)%5];
        return x+""+y;
    }
    if (vta.charAt(1)==vtb.charAt(1))
    {
        x=pf[((vta.charAt(0)-'0')+1)%5][(vta.charAt(1)-'0')];
        y=pf[((vtb.charAt(0)-'0')+1)%5][(vtb.charAt(1)-'0')];
        return x+""+y;
    }
    x=pf[vta.charAt(0)-'0'][vtb.charAt(1)-'0'];
    y=pf[vtb.charAt(0)-'0'][vta.charAt(1)-'0'];
    return x+""+y;
}
```

```
private String FindLocation(char a)
{
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            if (pf[i][j]==a) {
                return i+""+j;
            }
        }
    }
    return "";
}
```

```
private String Decrypt(String banma)
{
    int n=banma.length();
    String banro="";
    char a,b;
    for (int i = 0; i < n; i+=2) {
        a=banma.charAt(i);
        b=banma.charAt(i+1);
        banro+=ReverseReplace(a,b);
    }
    return banro;
}

String ReverseReplace(char a,char b){
    String vta=FindLocation(a);
    String vtb=FindLocation(b);
    char x,y;
    if (vta.charAt(0)==vtb.charAt(0)) {
        x=pf[vta.charAt(0)-'0'][((vta.charAt(1)-'0')-1+5)%5];
        y=pf[vtb.charAt(0)-'0'][((vtb.charAt(1)-'0')-1+5)%5];
        return x+""+y;
    }
    if (vta.charAt(1)==vtb.charAt(1)) {
        x=pf[((vta.charAt(0)-'0')-1+5)%5][(vta.charAt(1)-'0')];
        y=pf[((vtb.charAt(0)-'0')-1+5)%5][(vtb.charAt(1)-'0')];
        return x+""+y;
    }
    x=pf[vta.charAt(0)-'0'][vtb.charAt(1)-'0'];
    y=pf[vtb.charAt(0)-'0'][vta.charAt(1)-'0'];
    return x+""+y;
}
```

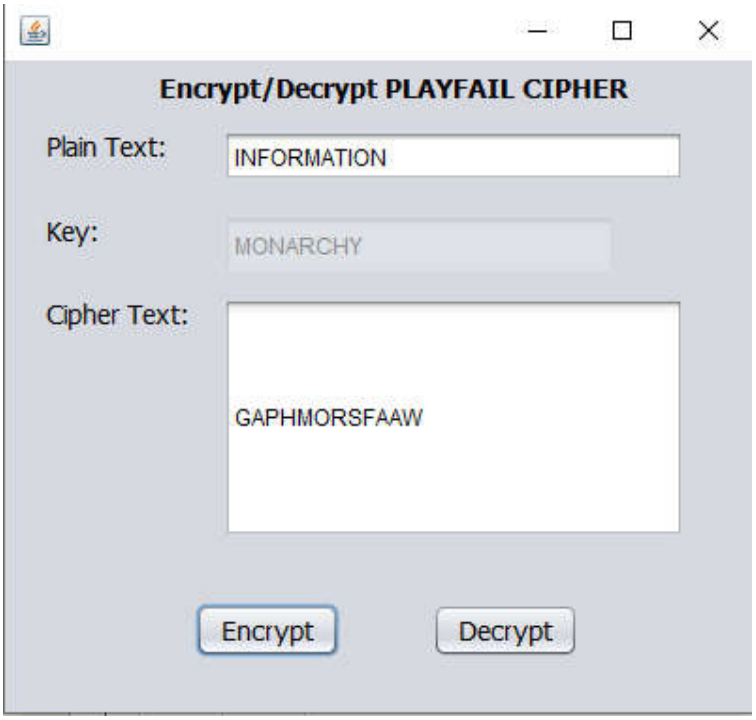
3.2 Button Encrypt:

```
private void btnEncryptActionPerformed(java.awt.event.ActionEvent evt) {
    String banro = this.txtPlainText.getText();
    banro= banro.toUpperCase();
    banro=banro.replace('J', 'I');
    String banma=Encrypt(banro);
    this.txtCipherText.setText(banma);

}
```

3.3 Button Decrypt:

```
private void btnDecryptActionPerformed(java.awt.event.ActionEvent evt) {

    String banma=this.txtCipherText.getText();
    String banro=Decrypt(banma);
    int n = banro.length();
    String br="";
    for (int i = 0; i < n-2; i+=2) {
        if (banro.charAt(i)==banro.charAt(i+2)) {
            br+=banro.charAt(i);
        }
        else{
            br+=banro.charAt(i)+""+banro.charAt(i+1);
        }
    }
    if (banro.charAt(n-1)=='X') {
        br+=banro.charAt(n-2);
    }
    else
        br+=banro.charAt(n-2);br+=banro.charAt(n-1);
    this.txtPlainText.setText(br);
}
```

Result:

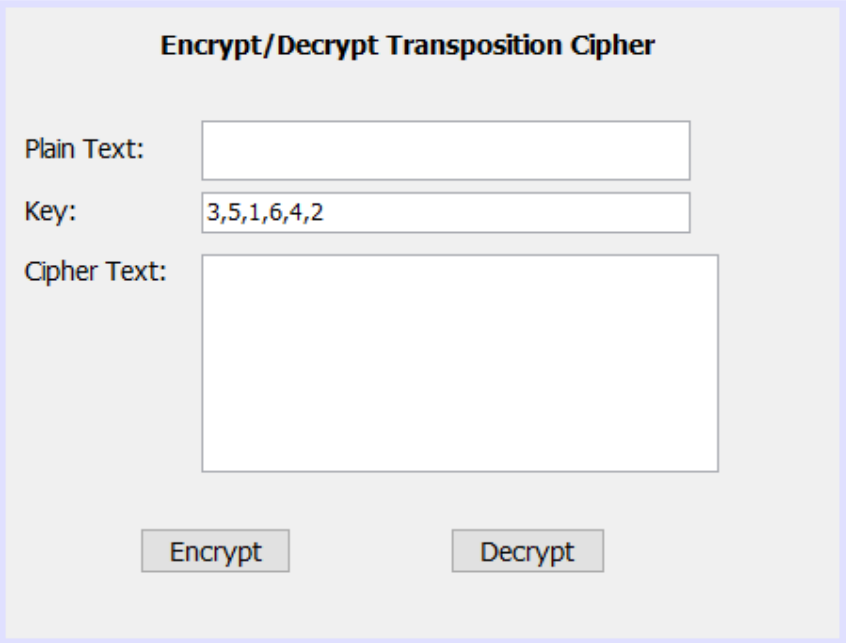


3.2.3 Transposition cipher

Write a program to encrypt and decrypt text with Playfail encryption algorithm. The program can perform the following functions:

- Allow text input into the system.
- Allows entering text protection keys.

❖ **Step 1: Design Form:**



❖ **Step 2: Write an event handler function:**

2.1 Button Encrypt:

```
private void btnEncryptActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String k = this.txtKey.getText();
    String ks[]= new String[6];
    ks=k.split(",");
    int key[]=new int[6];
    for (int i = 0; i < 6; i++) {
        key[i]=Integer.valueOf(ks[i])-1;
    }
    String sa=this.txtPlainText.getText();
    String kq="";
    int na=sa.length();
    int d=0;
    int c;
    String s="";
    int thieu=6-na%6;
    for (int i = 0; i < thieu; i++) {
        sa=sa+" ";
    }
    while(d<na)
    {
        c=d+6;
        s=sa.substring(d,c);
        for (int i = 0; i < 6; i++) {
            kq=kq+s.charAt(key[i]);
        }
        d=d+6;
    }
    this.txtCipherText.setText(kq);
}
```

2.2 Button Decrypt:

```
private void btnDecryptActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String k=this.txtKey.getText();
    String ks[]=new String[6];
    ks=k.split(",");
    int key[]=new int[6];
    for (int i = 0; i < 6; i++) {
        key[i]=Integer.valueOf(ks[i])-1;
    }
    int keytl[]=new int[6];
    for (int i = 0; i < 6; i++) {
        keytl[key[i]]=i;
    }
    String sa=this.txtCipherText.getText();
    String kq="";
    int na=sa.length();
    int d=0;
    int c;
    String s="";
    while (d<na) {
        c=d+6;
        s=sa.substring(d,c);
        for (int i = 0; i < 6; i++) {
            kq=kq+s.charAt(keytl[i]);
        }
        d=d+6;
    }
    this.txtPlainText.setText(kq);
}
```

Result: