TRIPLE DES ENCRYPTION AND DECRYPTION WITH ELECTRONIC CODE BOOK (ECB) MODE OF OPERATION

CPIT 425 Report

By

Abdulaziz Adnan Alsharif	2036023
Rakan Adnan Salama	2037276
Fahad Hamad Alsifri	1743998
Omar Saeed Al-Zahrani	2040569
Nasser Abdulrahman Alharbi	2037675

Supervised By

Dr. Iftikhar Ahmad

Department of Information Technology

Faculty of Computing and Information Technology

King Abdulaziz University

Jeddah – Saudi Arabia

[Fall 2023]

```
package pkg425project;

public class menu extends javax.swing.JFrame {

    public menu() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("PROJECT 425");

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel2.setText("TRIPLE DES ENCRYPTION AND DECRYPTION WITH");

        jButton1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jButton1.setText("TWO KEYS");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jButton2.setText("THREE KEYS");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
        jLabel3.setText("2036023  Abdulaziz Adnan Alsharif");

        jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel4.setText("2037276  Rakan Adnan Salama");

        jLabel5.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel5.setText("2040569  Omar Saeed Alzahrani");

        jLabel6.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel6.setText("1743998  Fahad Hamad Alsifri");

        jLabel7.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel7.setText("2037675  Nasser Abdulrahman Alharbi");

        jLabel9.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel9.setText("build by:");

        jLabel10.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel10.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel10.setText("build for:");

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel11.setText("Dr. Iftikhar Ahmad ");

        jLabel12.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel12.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel12.setText("CHOOSE HOW MANY KEY");

        jLabel13.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel13.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel13.setText("ELECTRONIC CODE BOOK (ECB) MODE OF OPERATION");

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(jLabel10, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
                    .addComponent(jLabel11,
javax.swing.GroupLayout.Alignment.TRAILING,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel7,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel6,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel5,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel4,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel9,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 225,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 225,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel3,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel12,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel13,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                    .addContainerGap())
            );
            layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
                .addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel9)
                .addGap(4, 4, 4)
                .addComponent(jLabel3)
                .addGap(4, 4, 4)
                .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel12)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G, false)
                    .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE, 125, Short.MAX_VALUE)
                    .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addContainerGap())
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
                Three Three = new Three();
        Three.show();
        dispose();
    }//GEN-LAST:event_jButton2ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
                        Two Two = new Two();
        Two.show();
        dispose();
    }//GEN-LAST:event_jButton1ActionPerformed
```

```
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(menu.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(menu.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(menu.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(menu.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new menu().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel2;
```

```
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel9;
    // End of variables declaration//GEN-END:variables
}
```

```
package pkg425project;

import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author gg
 */
public class Two extends javax.swing.JFrame {

    /**
     * Creates new form Two
     */
    public Two() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jTextField3 = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        jTextField4 = new javax.swing.JTextField();
```

```
        jButton1 = new javax.swing.JButton();
        jComboBox1 = new javax.swing.JComboBox<>();
        jComboBox3 = new javax.swing.JComboBox<>();
        jComboBox7 = new javax.swing.JComboBox<>();
        jComboBox8 = new javax.swing.JComboBox<>();
        jPanel2 = new javax.swing.JPanel();
        jLabel11 = new javax.swing.JLabel();
        jTextField5 = new javax.swing.JTextField();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jTextField6 = new javax.swing.JTextField();
        jLabel14 = new javax.swing.JLabel();
        jTextField7 = new javax.swing.JTextField();
        jLabel15 = new javax.swing.JLabel();
        jTextField8 = new javax.swing.JTextField();
        jButton3 = new javax.swing.JButton();
        jComboBox2 = new javax.swing.JComboBox<>();
        jComboBox4 = new javax.swing.JComboBox<>();
        jComboBox5 = new javax.swing.JComboBox<>();
        jComboBox6 = new javax.swing.JComboBox<>();
        jButton4 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("ENCRYPT");

        jLabel3.setText("PLAIN TEXT");

        jLabel4.setText("KEY ONE");

        jLabel5.setText("KEY TWO");

        jLabel6.setText("ENCRYPTED TEXT");

        jButton1.setText("ENCRYPT");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));
        jComboBox1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox1ActionPerformed(evt);
            }
        });

        jComboBox3.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "HEXADECIMAL", "BINARY" }));
```

```
        jComboBox7.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jComboBox8.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                    .addComponent(jButton1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.TRAILING, false)
                            .addComponent(jTextField1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 299, Short.MAX_VALUE)
                            .addComponent(jTextField2,
javax.swing.GroupLayout.Alignment.LEADING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
                            .addComponent(jComboBox7, 0, 150,
Short.MAX_VALUE)
                            .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                            .addComponent(jLabel6)
                            .addComponent(jLabel4)
                            .addComponent(jLabel5)
                            .addComponent(jLabel3))
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.TRAILING, false)
```

```
                                .addComponent(jTextField4,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 299, Short.MAX_VALUE)
                                .addComponent(jTextField3,
javax.swing.GroupLayout.Alignment.LEADING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
                                .addComponent(jComboBox8, 0, 150,
Short.MAX_VALUE)
                                .addComponent(jComboBox3, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel3)
                .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel4)
                .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox7,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel5)
                .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField3,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox8,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel6)
                .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel11.setText("DECRYPT");

        jLabel12.setText("PLAIN TEXT");

        jLabel13.setText("KEY ONE");

        jLabel14.setText("KEY TWO");

        jLabel15.setText("DECRYPTED TEXT");

        jButton3.setText("DECRYPT");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "HEXADECIMAL", "BINARY" }));

        jComboBox4.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jComboBox5.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));
```

```
        jComboBox6.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                    .addComponent(jButton3,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel11,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)

.addGroup(jPanel2Layout.createSequentialGroup()
                                    .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(jComboBox2, 0, 150,
Short.MAX_VALUE))
                                .addComponent(jLabel15)
                                .addComponent(jLabel13)
                                .addComponent(jLabel14)
                                .addComponent(jLabel12)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
                                        .addComponent(jTextField6,
javax.swing.GroupLayout.DEFAULT_SIZE, 300, Short.MAX_VALUE)
                                        .addComponent(jTextField7))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
                                        .addComponent(jComboBox5, 0, 150,
```

```
Short.MAX_VALUE)
                                            .addComponent(jComboBox6, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                                .addGroup(jPanel2Layout.createSequentialGroup()
                                    .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                            .addGap(0, 0, Short.MAX_VALUE)))
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel12)
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel13)
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox5,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel14)
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                        .addComponent(jComboBox6,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel15)
                    .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addContainerGap())
            );

            jButton4.setText("BACK TO MENU");
            jButton4.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    jButton4ActionPerformed(evt);
                }
            });

            javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
            getContentPane().setLayout(layout);
            layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
                        .addComponent(jButton4,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 476,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
                    .addContainerGap())
            );
            layout.setVerticalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G, false)
                    .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
44, Short.MAX_VALUE)
                .addContainerGap())
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents
String Efactore(String x, int y) {
        if (y == 0) {
            if (jComboBox1.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox1.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 1) {
            if (jComboBox7.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox7.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 2) {
            if (jComboBox8.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox8.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 3) {
            if (jComboBox3.getSelectedIndex() == 0) {
                return x;
            } else if (jComboBox3.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        }
        return x;

    }

    String Dfactore(String x, int y) {
        if (y == 0) {
            if (jComboBox2.getSelectedIndex() == 0) {
                return hexToBin(x);
            } else if (jComboBox2.getSelectedIndex() == 1) {
                return x;
```

```
            }
        } else if (y == 1) {
            if (jComboBox5.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox5.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 2) {
            if (jComboBox6.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox6.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 3) {
            if (jComboBox4.getSelectedIndex() == 0) {
                return hexToText(x);
            } else if (jComboBox4.getSelectedIndex() == 1) {
                return hexToBin(x);
            } else {
                return x;
            }
        }
        return x;

    }

    public static boolean isValidHex(String s) {
        char[] chars = s.toCharArray();
        for (char c : chars) {
            if (!Character.isDigit(c) && !(c >= 'a' && c <= 'f') && !(c >=
'A' && c <= 'F')) {
                return false;
            }
        }
        return true;
    }

    public static boolean isValidBinary(String s) {
        char[] chars = s.toCharArray();
        for (char c : chars) {
            if (c != '0' && c != '1') {
                return false;
            }
        }
        return true;
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
        boolean check = false;
        boolean check1 = false;
        if (jComboBox7.getSelectedIndex() == 0) {
            if (jTextField2.getText().length() == 8) {
                check = true;
            }
        } else if (jComboBox7.getSelectedIndex() == 1) {
            if (jTextField2.getText().length() == 16) {
```

```
                check = true;
                if (isValidHex(jTextField2.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        } else if (jComboBox7.getSelectedIndex() == 1) {
            if (jTextField2.getText().length() == 64) {
                check = true;
                if (isValidBinary(jTextField2.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        }

        if (jComboBox8.getSelectedIndex() == 0) {
            if (jTextField3.getText().length() == 8) {
                check1 = true;
            }
        } else if (jComboBox8.getSelectedIndex() == 1) {
            if (jTextField3.getText().length() == 16) {
                check1 = true;
                if (isValidHex(jTextField3.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        } else if (jComboBox8.getSelectedIndex() == 1) {
            if (jTextField3.getText().length() == 64) {
                check1 = true;
                if (isValidBinary(jTextField3.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        }

        if (check && check1) {
            if (jComboBox1.getSelectedIndex() == 0) {
                if ((jTextField1.getText().length() % 8 == 0)) {
                    if (jTextField1.getText().length() == 8) {
                        String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                        String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                        String binaryNumber0 =
Efactore(jTextField1.getText(), 0);
                        String binaryNumber1 = hexToBin(des(binaryNumber0,
```

```
keyArray1));
                        String binaryNumber2 = hexToBin(des1(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des(binaryNumber2, keyArray1);
                        jTextField4.setText(Efactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < Efactore(jTextField1.getText(),
0).length(); i += groupSize) {
                            int endIndex = Math.min(i + groupSize,
Efactore(jTextField1.getText(), 0).length());
                            String group = Efactore(jTextField1.getText(),
0).substring(i, endIndex);
                            String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                            String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des1(binaryNumber1, keyArray2));
                            String binaryNumber3 = des(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;

                        }

                        jTextField4.setText(Efactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 8, CHECK PLAIN TEXT");
                }

            } else if (jComboBox1.getSelectedIndex() == 1) {
                if ((jTextField1.getText().length() % 16 == 0)) {
                    if (jTextField1.getText().length() == 16) {
                        String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                        String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                        String binaryNumber0 =
Efactore(jTextField1.getText(), 0);
                        String binaryNumber1 = hexToBin(des(binaryNumber0,
keyArray1));
                        String binaryNumber2 = hexToBin(des1(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des(binaryNumber2, keyArray1);
                        jTextField4.setText(Efactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < Efactore(jTextField1.getText(),
0).length(); i += groupSize) {
                            int endIndex = Math.min(i + groupSize,
```

```
Efactore(jTextField1.getText(), 0).length());
                            String group = Efactore(jTextField1.getText(),
0).substring(i, endIndex);
                            String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                            String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des1(binaryNumber1, keyArray2));
                            String binaryNumber3 = des(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;

                        }
                        jTextField4.setText(Efactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 16, CHECK PLAIN TEXT");
                }

            } else if (jComboBox1.getSelectedIndex() == 2) {
                if ((jTextField1.getText().length() % 64 == 0)) {
                    if (jTextField1.getText().length() == 64) {
                        String[] keyArray1 = keyArray(jTextField2.getText());
                        String[] keyArray2 = keyArray(jTextField3.getText());
                        String binaryNumber0 = jTextField1.getText();
                        String binaryNumber1 = hexToBin(des(binaryNumber0,
keyArray1));
                        String binaryNumber2 = hexToBin(des1(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des(binaryNumber2, keyArray1);
                        jTextField4.setText(Efactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < jTextField1.getText().length(); i
+= groupSize) {
                            int endIndex = Math.min(i + groupSize,
jTextField1.getText().length());
                            String group = jTextField1.getText().substring(i,
endIndex);
                            String[] keyArray1 =
keyArray(jTextField2.getText());
                            String[] keyArray2 =
keyArray(jTextField3.getText());
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des1(binaryNumber1, keyArray2));
                            String binaryNumber3 = des(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;
```

```
                        }
                        jTextField4.setText(Efactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 64, CHECK PLAIN TEXT");
                }
            }

        } else {
            JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF THE
INPUTS OF THE KEYS AND THE INPUT OF THE KEY FOR THE CHOSEN REPRESENTATION");
        }
    }//GEN-LAST:event_jButton1ActionPerformed

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton3ActionPerformed
        boolean check = false;
        boolean check1 = false;
        if (jComboBox5.getSelectedIndex() == 0) {
            if (jTextField6.getText().length() == 8) {
                check = true;
            }
        } else if (jComboBox5.getSelectedIndex() == 1) {
            if (jTextField6.getText().length() == 16) {
                check = true;
                if (isValidHex(jTextField6.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        } else if (jComboBox5.getSelectedIndex() == 1) {
            if (jTextField6.getText().length() == 64) {
                check = true;
                if (isValidBinary(jTextField6.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        }

        if (jComboBox6.getSelectedIndex() == 0) {
            if (jTextField7.getText().length() == 8) {
                check1 = true;
            }
        } else if (jComboBox6.getSelectedIndex() == 1) {
            if (jTextField7.getText().length() == 16) {
                check1 = true;
                if (isValidHex(jTextField7.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
```

```
                }
            }

        } else if (jComboBox6.getSelectedIndex() == 1) {
            if (jTextField7.getText().length() == 64) {
                check1 = true;
                if (isValidBinary(jTextField7.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        }
        if (check && check1) {

            if (jComboBox2.getSelectedIndex() == 0) {
                if ((jTextField5.getText().length() % 16 == 0)) {
                    if (jTextField5.getText().length() == 16) {
                        String[] keyArray1 =
keyArray(Dfactore(jTextField6.getText(), 1));
                        String[] keyArray2 =
keyArray(Dfactore(jTextField7.getText(), 2));
                        String binaryNumber0 =
Dfactore(jTextField5.getText(), 0);
                        String binaryNumber1 = hexToBin(des1(binaryNumber0,
keyArray1));
                        String binaryNumber2 = hexToBin(des(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                        jTextField8.setText(Dfactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < Dfactore(jTextField5.getText(),
0).length(); i += groupSize) {
                            int endIndex = Math.min(i + groupSize,
Dfactore(jTextField5.getText(), 0).length());
                            String group = Dfactore(jTextField5.getText(),
0).substring(i, endIndex);
                            String[] keyArray1 =
keyArray(Dfactore(jTextField6.getText(), 1));
                            String[] keyArray2 =
keyArray(Dfactore(jTextField7.getText(), 2));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des1(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des(binaryNumber1, keyArray2));
                            String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;

                        }
                        jTextField8.setText(Dfactore(binaryNumber4, 3));
                    }
```

```
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 16, CHECK PLAIN TEXT");
                }

            } else if (jComboBox2.getSelectedIndex() == 1) {
                if ((jTextField5.getText().length() % 64 == 0)) {
                    if (jTextField5.getText().length() == 64) {
                        String[] keyArray1 = keyArray(jTextField6.getText());
                        String[] keyArray2 = keyArray(jTextField7.getText());
                        String binaryNumber0 = jTextField5.getText();
                        String binaryNumber1 = hexToBin(des1(binaryNumber0,
keyArray1));
                        String binaryNumber2 = hexToBin(des(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                        jTextField8.setText(Dfactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < jTextField5.getText().length(); i
+= groupSize) {
                            int endIndex = Math.min(i + groupSize,
jTextField5.getText().length());
                            String group = jTextField5.getText().substring(i,
endIndex);
                            String[] keyArray1 =
keyArray(jTextField6.getText());
                            String[] keyArray2 =
keyArray(jTextField7.getText());
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des1(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des(binaryNumber1, keyArray2));
                            String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;

                        }
                        jTextField8.setText(Dfactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 64, CHECK PLAIN TEXT");
                }
            }

        } else {
            JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF THE
INPUTS OF THE KEYS AND THE INPUT OF THE KEY FOR THE CHOSEN REPRESENTATION");
        }

    }//GEN-LAST:event_jButton3ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
```

```
{//GEN-FIRST:event_jButton4ActionPerformed
        menu menu = new menu();
        menu.show();
        dispose();
    }//GEN-LAST:event_jButton4ActionPerformed

    private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jComboBox1ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_jComboBox1ActionPerformed
    public static String textToBinary(String text) {
        byte[] bytes = text.getBytes();
        StringBuilder binaryStringBuilder = new StringBuilder();
        for (byte b : bytes) {
            int value = b;
            for (int i = 7; i >= 0; i--) {
                binaryStringBuilder.append((value & (1 << i)) == 0 ? '0' :
'1');
            }
            // Add a space between bytes for readability
        }
        return binaryStringBuilder.toString();
    }

    public static String hexToText(String hex) {
        byte[] bytes = new byte[hex.length() / 2];
        for (int i = 0; i < hex.length(); i += 2) {
            int byteValue = Integer.parseInt(hex.substring(i, i + 2), 16);
            bytes[i / 2] = (byte) byteValue;
        }

        try {
            return new String(bytes, "UTF-8");
        } catch (UnsupportedEncodingException ex) {
            Logger.getLogger(Two.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return null;

    }

    public static String des(String binaryString, String[] keys) {
        System.out.println("first " + binaryString);
        // Define the initial permutation table for DES
        int[] initialPermutationTable = {
            58, 50, 42, 34, 26, 18, 10, 2,
            60, 52, 44, 36, 28, 20, 12, 4,
            62, 54, 46, 38, 30, 22, 14, 6,
            64, 56, 48, 40, 32, 24, 16, 8,
            57, 49, 41, 33, 25, 17, 9, 1,
            59, 51, 43, 35, 27, 19, 11, 3,
            61, 53, 45, 37, 29, 21, 13, 5,
            63, 55, 47, 39, 31, 23, 15, 7
        };

        // Perform the initial permutation
        StringBuilder ipresult = new StringBuilder();
```

```
        for (int i : initialPermutationTable) {
            ipresult.append(binaryString.charAt(i - 1));
        }
        System.out.println("after ip " + ipresult);
        String[][] ipArrayResult = new String[17][2];
        StringBuilder sbresultl = new StringBuilder();
        StringBuilder sbresultr = new StringBuilder();
        for (int j = 0; j < 64; j++) {

            if (j < 32) {
                sbresultl.append(ipresult.charAt(j));
            }
            if (j >= 32) {
                sbresultr.append(ipresult.charAt(j));
            }
            ipArrayResult[0][0] = sbresultl.toString();
            ipArrayResult[0][1] = sbresultr.toString();
        }
        System.out.println("l0  " + ipArrayResult[0][0]);
        System.out.println("r0 " + ipArrayResult[0][1]);
        for (int i = 1; i < 17; i++) {

            ipArrayResult[i][0] = ipArrayResult[i - 1][1];
            System.out.println("l" + i + " " + ipArrayResult[i][0]);
            int[] eBitSelectionTable = {
                32, 1, 2, 3, 4, 5,
                4, 5, 6, 7, 8, 9,
                8, 9, 10, 11, 12, 13,
                12, 13, 14, 15, 16, 17,
                16, 17, 18, 19, 20, 21,
                20, 21, 22, 23, 24, 25,
                24, 25, 26, 27, 28, 29,
                28, 29, 30, 31, 32, 1
            };
            StringBuilder eresult = new StringBuilder();
            for (int j : eBitSelectionTable) {
                eresult.append(ipArrayResult[i - 1][1].charAt(j - 1));
            }
            //0110111111
            System.out.println("rebit " + eresult);
            System.out.println("key" + (i - 1) + " " + keys[i - 1]);
            StringBuilder xourEKresult = new StringBuilder();
            for (int j = 0; j < eresult.length(); j++) {
                xourEKresult.append((char) ('0' + (eresult.charAt(j) ^ keys[i
- 1].charAt(j))));
            }
            System.out.println("xorEbitKey " + xourEKresult);

            String[] sboxArray = new String[8];
            int count = 0;
            StringBuilder ssresult = new StringBuilder();
            for (int j = 0; j < 48; j++) {
                if (j % 6 == 0 && j > 0) {
                    sboxArray[count++] = ssresult.toString();
                    ssresult = new StringBuilder();
                }
                ssresult.append(xourEKresult.charAt(j));
```

```
            }
            if (ssresult.length() > 0) {
                sboxArray[count] = ssresult.toString();
            }

            StringBuilder sboxsresult = new StringBuilder();
            int[][] s1Box = {
                {14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7},
                {0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8},
                {4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0},
                {15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13}
            };
            String x = String.valueOf(sboxArray[0].charAt(0) + "" +
sboxArray[0].charAt(sboxArray[0].length() - 1));
            String y = String.valueOf(sboxArray[0].substring(1, 5));
            int row = Integer.parseInt(x, 2);
            int colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s1Box[row][colm])).replace(' ', '0'));

            int[][] s2Box = {
                {15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10},
                {3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5},
                {0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15},
                {13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9}
            };
            x = String.valueOf(sboxArray[1].charAt(0) + "" +
sboxArray[1].charAt(sboxArray[1].length() - 1));
            y = String.valueOf(sboxArray[1].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s2Box[row][colm])).replace(' ', '0'));
            int[][] s3Box = {
                {10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8},
                {13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1},
                {13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7},
                {1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12}
            };
            x = String.valueOf(sboxArray[2].charAt(0) + "" +
sboxArray[2].charAt(sboxArray[2].length() - 1));
            y = String.valueOf(sboxArray[2].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s3Box[row][colm])).replace(' ', '0'));
            if (i == 6) {

                System.out.println(sboxsresult.charAt(6));
            }
            int[][] s4Box = {
                {7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15},
                {13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9},
                {10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4},
                {3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14}
            };
            x = String.valueOf(sboxArray[3].charAt(0) + "" +
```

```
sboxArray[3].charAt(sboxArray[3].length() - 1));
            y = String.valueOf(sboxArray[3].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s4Box[row][colm])).replace(' ', '0'));
            int[][] s5Box = {
                {2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9},
                {14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6},
                {4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14},
                {11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3}
            };
            x = String.valueOf(sboxArray[4].charAt(0) + "" +
sboxArray[4].charAt(sboxArray[4].length() - 1));
            y = String.valueOf(sboxArray[4].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s5Box[row][colm])).replace(' ', '0'));
            int[][] s6Box = {
                {12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11},
                {10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8},
                {9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6},
                {4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13}
            };
            x = String.valueOf(sboxArray[5].charAt(0) + "" +
sboxArray[5].charAt(sboxArray[5].length() - 1));
            y = String.valueOf(sboxArray[5].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s6Box[row][colm])).replace(' ', '0'));

            int[][] s7Box = {
                {4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1},
                {13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6},
                {1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2},
                {6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12}
            };
            x = String.valueOf(sboxArray[6].charAt(0) + "" +
sboxArray[6].charAt(sboxArray[6].length() - 1));
            y = String.valueOf(sboxArray[6].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s7Box[row][colm])).replace(' ', '0'));

            int[][] s8Box = {
                {13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7},
                {1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2},
                {7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8},
                {2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11}
            };
            x = String.valueOf(sboxArray[7].charAt(0) + "" +
sboxArray[7].charAt(sboxArray[7].length() - 1));
            y = String.valueOf(sboxArray[7].substring(1, 5));
            row = Integer.parseInt(x, 2);
```

```
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s8Box[row][colm])).replace(' ', '0'));
            System.out.println("sbox " + sboxsresult);

            int[] pPermutationTable = {
                16, 7, 20, 21,
                29, 12, 28, 17,
                1, 15, 23, 26,
                5, 18, 31, 10,
                2, 8, 24, 14,
                32, 27, 3, 9,
                19, 13, 30, 6,
                22, 11, 4, 25
            };

            StringBuilder pPermutat = new StringBuilder();
            for (int j : pPermutationTable) {
                pPermutat.append(sboxsresult.charAt(j - 1));
            }
            System.out.println("pbox " + pPermutat);
            System.out.println("left " + (i - 1) + " " + ipArrayResult[i -
1][0]);
            StringBuilder xourplresult = new StringBuilder();
            for (int j = 0; j < pPermutat.length(); j++) {
                xourplresult.append((char) ('0' + (pPermutat.charAt(j) ^
ipArrayResult[i - 1][0].charAt(j))));
            }
            System.out.println("pbox " + xourplresult);
            ipArrayResult[i][1] = xourplresult.toString();
            System.out.println("r " + i + " " + xourplresult);
            // keyArray[i]=eresult.toString();
            if (i == 2) {
                System.out.println();

            }
        }
        System.out.println("end");
        String x = ipArrayResult[16][0];
        System.out.println("l " + x);
        String y = ipArrayResult[16][1];
        System.out.println("r " + y);
        String z = y + "" + x;
        System.out.println("beforinv " + z);
        int[] invip1PermutationTable = {
            40, 8, 48, 16, 56, 24, 64, 32,
            39, 7, 47, 15, 55, 23, 63, 31,
            38, 6, 46, 14, 54, 22, 62, 30,
            37, 5, 45, 13, 53, 21, 61, 29,
            36, 4, 44, 12, 52, 20, 60, 28,
            35, 3, 43, 11, 51, 19, 59, 27,
            34, 2, 42, 10, 50, 18, 58, 26,
            33, 1, 41, 9, 49, 17, 57, 25
        };
        StringBuilder invPermutat = new StringBuilder();
        for (int j : invip1PermutationTable) {
            invPermutat.append(z.charAt(j - 1));
```

```
        }
        System.out.println("afterinv " + invPermutat);

        BigInteger decimal = new BigInteger(String.valueOf(invPermutat), 2);

        // Convert decimal to hexadecimal
        String hexString = decimal.toString(16);
        System.out.println("to hex " + hexString);
        return hexString;
    }

    public static String des1(String binaryString, String[] keys) {
        System.out.println("first " + binaryString);
        // Define the initial permutation table for DES
        int[] initialPermutationTable = {
            58, 50, 42, 34, 26, 18, 10, 2,
            60, 52, 44, 36, 28, 20, 12, 4,
            62, 54, 46, 38, 30, 22, 14, 6,
            64, 56, 48, 40, 32, 24, 16, 8,
            57, 49, 41, 33, 25, 17, 9, 1,
            59, 51, 43, 35, 27, 19, 11, 3,
            61, 53, 45, 37, 29, 21, 13, 5,
            63, 55, 47, 39, 31, 23, 15, 7
        };

        // Perform the initial permutation
        StringBuilder ipresult = new StringBuilder();
        for (int i : initialPermutationTable) {
            ipresult.append(binaryString.charAt(i - 1));
        }
        System.out.println("after ip " + ipresult);
        String[][] ipArrayResult = new String[17][2];
        StringBuilder sbresultl = new StringBuilder();
        StringBuilder sbresultr = new StringBuilder();
        for (int j = 0; j < 64; j++) {

            if (j < 32) {
                sbresultl.append(ipresult.charAt(j));
            }
            if (j >= 32) {
                sbresultr.append(ipresult.charAt(j));
            }
            ipArrayResult[0][0] = sbresultl.toString();
            ipArrayResult[0][1] = sbresultr.toString();
        }
        System.out.println("l0  " + ipArrayResult[0][0]);
        System.out.println("r0 " + ipArrayResult[0][1]);
        for (int i = 1; i < 17; i++) {

            ipArrayResult[i][0] = ipArrayResult[i - 1][1];
            System.out.println("l" + i + " " + ipArrayResult[i][0]);
            int[] eBitSelectionTable = {
                32, 1, 2, 3, 4, 5,
                4, 5, 6, 7, 8, 9,
                8, 9, 10, 11, 12, 13,
                12, 13, 14, 15, 16, 17,
                16, 17, 18, 19, 20, 21,
```

```
                20, 21, 22, 23, 24, 25,
                24, 25, 26, 27, 28, 29,
                28, 29, 30, 31, 32, 1
            };
            StringBuilder eresult = new StringBuilder();
            for (int j : eBitSelectionTable) {
                eresult.append(ipArrayResult[i - 1][1].charAt(j - 1));
            }
            //0110111111
            System.out.println("rebit " + eresult);
            System.out.println("key" + (i - 1) + " " + keys[(16 - i)]);
            StringBuilder xourEKresult = new StringBuilder();
            for (int j = 0; j < eresult.length(); j++) {
                xourEKresult.append((char) ('0' + (eresult.charAt(j) ^
keys[(16 - i)].charAt(j))));
            }
            System.out.println("xorEbitKey " + xourEKresult);

            String[] sboxArray = new String[8];
            int count = 0;
            StringBuilder ssresult = new StringBuilder();
            for (int j = 0; j < 48; j++) {
                if (j % 6 == 0 && j > 0) {
                    sboxArray[count++] = ssresult.toString();
                    ssresult = new StringBuilder();
                }
                ssresult.append(xourEKresult.charAt(j));
            }
            if (ssresult.length() > 0) {
                sboxArray[count] = ssresult.toString();
            }

            StringBuilder sboxsresult = new StringBuilder();
            int[][] s1Box = {
                {14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7},
                {0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8},
                {4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0},
                {15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13}
            };
            String x = String.valueOf(sboxArray[0].charAt(0) + "" +
sboxArray[0].charAt(sboxArray[0].length() - 1));
            String y = String.valueOf(sboxArray[0].substring(1, 5));
            int row = Integer.parseInt(x, 2);
            int colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s1Box[row][colm])).replace(' ', '0'));

            int[][] s2Box = {
                {15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10},
                {3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5},
                {0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15},
                {13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9}
            };
            x = String.valueOf(sboxArray[1].charAt(0) + "" +
sboxArray[1].charAt(sboxArray[1].length() - 1));
            y = String.valueOf(sboxArray[1].substring(1, 5));
            row = Integer.parseInt(x, 2);
```

```
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s2Box[row][colm])).replace(' ', '0'));
            int[][] s3Box = {
                {10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8},
                {13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1},
                {13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7},
                {1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12}
            };
            x = String.valueOf(sboxArray[2].charAt(0) + "" +
sboxArray[2].charAt(sboxArray[2].length() - 1));
            y = String.valueOf(sboxArray[2].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s3Box[row][colm])).replace(' ', '0'));
            if (i == 6) {

                System.out.println(sboxsresult.charAt(6));
            }
            int[][] s4Box = {
                {7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15},
                {13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9},
                {10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4},
                {3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14}
            };
            x = String.valueOf(sboxArray[3].charAt(0) + "" +
sboxArray[3].charAt(sboxArray[3].length() - 1));
            y = String.valueOf(sboxArray[3].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s4Box[row][colm])).replace(' ', '0'));
            int[][] s5Box = {
                {2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9},
                {14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6},
                {4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14},
                {11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3}
            };
            x = String.valueOf(sboxArray[4].charAt(0) + "" +
sboxArray[4].charAt(sboxArray[4].length() - 1));
            y = String.valueOf(sboxArray[4].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s5Box[row][colm])).replace(' ', '0'));
            int[][] s6Box = {
                {12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11},
                {10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8},
                {9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6},
                {4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13}
            };
            x = String.valueOf(sboxArray[5].charAt(0) + "" +
sboxArray[5].charAt(sboxArray[5].length() - 1));
            y = String.valueOf(sboxArray[5].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
```

```
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s6Box[row][colm])).replace(' ', '0'));

            int[][] s7Box = {
                {4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1},
                {13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6},
                {1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2},
                {6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12}
            };
            x = String.valueOf(sboxArray[6].charAt(0) + "" +
sboxArray[6].charAt(sboxArray[6].length() - 1));
            y = String.valueOf(sboxArray[6].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s7Box[row][colm])).replace(' ', '0'));

            int[][] s8Box = {
                {13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7},
                {1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2},
                {7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8},
                {2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11}
            };
            x = String.valueOf(sboxArray[7].charAt(0) + "" +
sboxArray[7].charAt(sboxArray[7].length() - 1));
            y = String.valueOf(sboxArray[7].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s8Box[row][colm])).replace(' ', '0'));
            System.out.println("sbox " + sboxsresult);

            int[] pPermutationTable = {
                16, 7, 20, 21,
                29, 12, 28, 17,
                1, 15, 23, 26,
                5, 18, 31, 10,
                2, 8, 24, 14,
                32, 27, 3, 9,
                19, 13, 30, 6,
                22, 11, 4, 25
            };

            StringBuilder pPermutat = new StringBuilder();
            for (int j : pPermutationTable) {
                pPermutat.append(sboxsresult.charAt(j - 1));
            }
            System.out.println("pbox " + pPermutat);
            System.out.println("left " + (i - 1) + " " + ipArrayResult[i -
1][0]);
            StringBuilder xourplresult = new StringBuilder();
            for (int j = 0; j < pPermutat.length(); j++) {
                xourplresult.append((char) ('0' + (pPermutat.charAt(j) ^
ipArrayResult[i - 1][0].charAt(j))));
            }
            System.out.println("pbox " + xourplresult);
            ipArrayResult[i][1] = xourplresult.toString();
```

```
            System.out.println("r " + i + " " + xourplresult);
            // keyArray[i]=eresult.toString();
            if (i == 2) {
                System.out.println();

            }
        }
        System.out.println("end");
        String x = ipArrayResult[16][0];
        System.out.println("l " + x);
        String y = ipArrayResult[16][1];
        System.out.println("r " + y);
        String z = y + "" + x;
        System.out.println("beforinv " + z);
        int[] invip1PermutationTable = {
            40, 8, 48, 16, 56, 24, 64, 32,
            39, 7, 47, 15, 55, 23, 63, 31,
            38, 6, 46, 14, 54, 22, 62, 30,
            37, 5, 45, 13, 53, 21, 61, 29,
            36, 4, 44, 12, 52, 20, 60, 28,
            35, 3, 43, 11, 51, 19, 59, 27,
            34, 2, 42, 10, 50, 18, 58, 26,
            33, 1, 41, 9, 49, 17, 57, 25
        };
        StringBuilder invPermutat = new StringBuilder();
        for (int j : invip1PermutationTable) {
            invPermutat.append(z.charAt(j - 1));
        }
        System.out.println("afterinv " + invPermutat);

        BigInteger decimal = new BigInteger(String.valueOf(invPermutat), 2);

        // Convert decimal to hexadecimal
        String hexString = decimal.toString(16);
        System.out.println("to hex " + hexString);
        return hexString;
    }

    public static String hexToBin(String hex) {
        StringBuilder binary = new StringBuilder();
        for (int i = 0; i < hex.length(); i++) {
            char hexDigit = hex.charAt(i);
            int decimalValue = Character.digit(hexDigit, 16);
            String binaryValue = Integer.toBinaryString(decimalValue);
            // Ensure each binary representation has 4 bits
            while (binaryValue.length() < 4) {
                binaryValue = "0" + binaryValue;
            }
            binary.append(binaryValue);
        }
        return binary.toString();
    }

    public static String binToHex(String binary) {
        StringBuilder hex = new StringBuilder();
        for (int i = 0; i < binary.length(); i += 4) {
            // Ensure that the substring has 4 bits, otherwise add zeros to
```

```
the left
            String binSub = binary.substring(i, Math.min(i + 4,
binary.length()));
            while (binSub.length() < 4) {
                binSub = "0" + binSub;
            }

            int decimalValue = Integer.parseInt(binSub, 2);
            String hexValue =
Integer.toHexString(decimalValue).toUpperCase();

            hex.append(hexValue);
        }
        return hex.toString();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Two.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Two.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Two.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Two.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        System.out.println(binToHex(textToBinary("bbbbbbbb")));
        System.out.println(binToHex(textToBinary("cccccccc")));
```

```
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Two().setVisible(true);
            }
        });
    }

    public static String[] keyArray(String theKey) {

        // Define the initial permutation table for DES
        int[] pc1Table = {
            57, 49, 41, 33, 25, 17, 9,
            1, 58, 50, 42, 34, 26, 18,
            10, 2, 59, 51, 43, 35, 27,
            19, 11, 3, 60, 52, 44, 36,
            63, 55, 47, 39, 31, 23, 15,
            7, 62, 54, 46, 38, 30, 22,
            14, 6, 61, 53, 45, 37, 29,
            21, 13, 5, 28, 20, 12, 4
        };

        // Perform the initial permutation
        StringBuilder pc1result = new StringBuilder();
        for (int i : pc1Table) {
            pc1result.append(theKey.charAt(i - 1));
        }
        String binaryString = pc1result.toString();

        if (binaryString.length() % 2 != 0) {
            throw new IllegalArgumentException("Binary string length must be
even.");
        }
        String[][] pc1ArrayResult = new String[17][2];
        StringBuilder sbresultl = new StringBuilder();
        StringBuilder sbresultr = new StringBuilder();
        for (int j = 0; j < 56; j++) {

            if (j < 28) {
                sbresultl.append(binaryString.charAt(j));
            }
            if (j >= 28) {
                sbresultr.append(binaryString.charAt(j));
            }
            pc1ArrayResult[0][0] = sbresultl.toString();
            pc1ArrayResult[0][1] = sbresultr.toString();
        }

        for (int i = 1; i < 17; i++) {
            if (i == 1 || i == 2 || i == 9 || i == 16) {
                pc1ArrayResult[i][0] = shiftLeft(pc1ArrayResult[i - 1][0],
1);
                pc1ArrayResult[i][1] = shiftLeft(pc1ArrayResult[i - 1][1],
1);
            } else {

                pc1ArrayResult[i][0] = shiftLeft(pc1ArrayResult[i - 1][0],
```

```
2);
                pc1ArrayResult[i][1] = shiftLeft(pc1ArrayResult[i - 1][1],
2);
            }
        }
        String[] keyArray = new String[16];

        int[] pc2Table = {
            14, 17, 11, 24, 1, 5,
            3, 28, 15, 6, 21, 10,
            23, 19, 12, 4, 26, 8,
            16, 7, 27, 20, 13, 2,
            41, 52, 31, 37, 47, 55,
            30, 40, 51, 45, 33, 48,
            44, 49, 39, 56, 34, 53,
            46, 42, 50, 36, 29, 32
        };

        for (int i = 0; i < 16; i++) {
            keyArray[i] = pc1ArrayResult[i + 1][0] + pc1ArrayResult[i +
1][1];
            StringBuilder pc2result = new StringBuilder();
            for (int j : pc2Table) {
                pc2result.append(keyArray[i].charAt(j - 1));
            }
            keyArray[i] = pc2result.toString();
        }
        return keyArray;
    }

    public static String shiftLeft(String binaryString, int positions) {
        // Convert the binary string to a char array for easy manipulation
        char[] charArray = binaryString.toCharArray();

        // Perform the left shift for the specified number of positions
        for (int shift = 0; shift < positions; shift++) {
            char firstChar = charArray[0];

            for (int i = 1; i < charArray.length; i++) {
                charArray[i - 1] = charArray[i];
            }

            charArray[charArray.length - 1] = firstChar;
        }

        // Convert the char array back to a string
        return new String(charArray);
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JComboBox<String> jComboBox3;
    private javax.swing.JComboBox<String> jComboBox4;
    private javax.swing.JComboBox<String> jComboBox5;
```

```
    private javax.swing.JComboBox<String> jComboBox6;
    private javax.swing.JComboBox<String> jComboBox7;
    private javax.swing.JComboBox<String> jComboBox8;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    // End of variables declaration//GEN-END:variables
}
```

```
package pkg425project;

import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class Three extends javax.swing.JFrame {


    public Three() {
        initComponents();
    }


    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
```

```
        jLabel4 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jTextField3 = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        jTextField4 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jComboBox1 = new javax.swing.JComboBox<>();
        jComboBox3 = new javax.swing.JComboBox<>();
        jComboBox7 = new javax.swing.JComboBox<>();
        jComboBox8 = new javax.swing.JComboBox<>();
        jTextField9 = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jComboBox9 = new javax.swing.JComboBox<>();
        jPanel2 = new javax.swing.JPanel();
        jLabel11 = new javax.swing.JLabel();
        jTextField5 = new javax.swing.JTextField();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jTextField6 = new javax.swing.JTextField();
        jLabel14 = new javax.swing.JLabel();
        jTextField7 = new javax.swing.JTextField();
        jLabel15 = new javax.swing.JLabel();
        jTextField8 = new javax.swing.JTextField();
        jButton3 = new javax.swing.JButton();
        jComboBox2 = new javax.swing.JComboBox<>();
        jComboBox4 = new javax.swing.JComboBox<>();
        jComboBox5 = new javax.swing.JComboBox<>();
        jComboBox6 = new javax.swing.JComboBox<>();
        jLabel8 = new javax.swing.JLabel();
        jTextField10 = new javax.swing.JTextField();
        jComboBox10 = new javax.swing.JComboBox<>();
        jButton4 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("ENCRYPT");

        jLabel3.setText("PLAIN TEXT");

        jLabel4.setText("KEY ONE");

        jLabel5.setText("KEY TWO");

        jLabel6.setText("ENCRYPTED TEXT");

        jButton1.setText("ENCRYPT");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
```

```
        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));
        jComboBox1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox1ActionPerformed(evt);
            }
        });

        jComboBox3.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "HEXADECIMAL", "BINARY" }));

        jComboBox7.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jComboBox8.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jLabel7.setText("KEY THREE");

        jComboBox9.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                    .addComponent(jButton1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.TRAILING, false)
                            .addComponent(jTextField1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 299, Short.MAX_VALUE)
                            .addComponent(jTextField2,
javax.swing.GroupLayout.Alignment.LEADING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
```

```
                                .addComponent(jComboBox7, 0, 150,
Short.MAX_VALUE)
                                .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, 299,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jComboBox8,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addComponent(jTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE, 299,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jComboBox9,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE, 299,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                                .addComponent(jLabel4)
                                .addComponent(jLabel5)
                                .addComponent(jLabel3)
                                .addComponent(jLabel7)
                                .addComponent(jLabel6))
                            .addGap(0, 0, Short.MAX_VALUE)))
                    .addContainerGap())
            );
            jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel3)
                    .addGap(7, 7, 7)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel4)
                    .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox7,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel5)
                    .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox8,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel7)
                    .addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox9,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jLabel11.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel11.setText("DECRYPT");

        jLabel12.setText("PLAIN TEXT");

        jLabel13.setText("KEY ONE");

        jLabel14.setText("KEY TWO");

        jLabel15.setText("DECRYPTED TEXT");

        jButton3.setText("DECRYPT");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "HEXADECIMAL", "BINARY" }));

        jComboBox4.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jComboBox5.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jComboBox6.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        jLabel8.setText("KEY THREE");

        jComboBox10.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "TEXT", "HEXADECIMAL", "BINARY" }));

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                    .addComponent(jButton3,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel11,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                            .addComponent(jLabel15)
                            .addGroup(jPanel2Layout.createSequentialGroup()
                                .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jTextField10,
javax.swing.GroupLayout.PREFERRED_SIZE, 299,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jComboBox10,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)

.addGroup(jPanel2Layout.createSequentialGroup()
                                    .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(jComboBox2, 0, 150,
Short.MAX_VALUE))
                                .addComponent(jLabel13)
```

```
                                    .addComponent(jLabel14)
                                    .addComponent(jLabel12)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
                                            .addComponent(jTextField6,
javax.swing.GroupLayout.DEFAULT_SIZE, 300, Short.MAX_VALUE)
                                            .addComponent(jTextField7))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
                                            .addComponent(jComboBox5, 0, 150,
Short.MAX_VALUE)
                                            .addComponent(jComboBox6, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                                .addComponent(jLabel8))
                            .addGap(0, 0, Short.MAX_VALUE)))
                    .addContainerGap())
            );
            jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel12)
                    .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel13)
                    .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jComboBox5,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
                .addComponent(jLabel14)
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox6,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel8)
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField10,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox10,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel15)
                .addGap(7, 7, 7)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jTextField8,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton3,
javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        jButton4.setText("BACK TO MENU");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
                    .addComponent(jButton4,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 476,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
                    .addContainerGap())
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G, false)
                    .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
44, Short.MAX_VALUE)
                    .addContainerGap())
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents
String Efactore(String x, int y) {
        if (y == 0) {
            if (jComboBox1.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox1.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 1) {
            if (jComboBox7.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox7.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
```

```
        } else if (y == 2) {
            if (jComboBox8.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox8.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 3) {
            if (jComboBox3.getSelectedIndex() == 0) {
                return x;
            } else if (jComboBox3.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 4) {
            if (jComboBox9.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox9.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        }
        return x;

    }

    String Dfactore(String x, int y) {
        if (y == 0) {
            if (jComboBox2.getSelectedIndex() == 0) {
                return hexToBin(x);
            } else if (jComboBox2.getSelectedIndex() == 1) {
                return x;
            }
        } else if (y == 1) {
            if (jComboBox5.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox5.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 2) {
            if (jComboBox6.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox6.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        } else if (y == 3) {
            if (jComboBox4.getSelectedIndex() == 0) {
                return hexToText(x);
            } else if (jComboBox4.getSelectedIndex() == 1) {
                return hexToBin(x);
            } else {
                return x;
            }
        } else if (y == 4) {
            if (jComboBox10.getSelectedIndex() == 0) {
                return textToBinary(x);
            } else if (jComboBox10.getSelectedIndex() == 1) {
                return hexToBin(x);
            }
        }
```

```
        return x;

    }

    public static boolean isValidHex(String s) {
        char[] chars = s.toCharArray();
        for (char c : chars) {
            if (!Character.isDigit(c) && !(c >= 'a' && c <= 'f') && !(c >=
'A' && c <= 'F')) {
                return false;
            }
        }
        return true;
    }

    public static boolean isValidBinary(String s) {
        char[] chars = s.toCharArray();
        for (char c : chars) {
            if (c != '0' && c != '1') {
                return false;
            }
        }
        return true;
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
        boolean check = false;
        boolean check1 = false;
        boolean check2 = false;
        if (jComboBox7.getSelectedIndex() == 0) {
            if (jTextField2.getText().length() == 8) {
                check = true;
            }
        } else if (jComboBox7.getSelectedIndex() == 1) {
            if (jTextField2.getText().length() == 16) {
                check = true;
                if (isValidHex(jTextField2.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        } else if (jComboBox7.getSelectedIndex() == 1) {
            if (jTextField2.getText().length() == 64) {
                check = true;
                if (isValidBinary(jTextField2.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        }

        if (jComboBox8.getSelectedIndex() == 0) {
```

```
            if (jTextField3.getText().length() == 8) {
                check1 = true;
            }
        } else if (jComboBox8.getSelectedIndex() == 1) {
            if (jTextField3.getText().length() == 16) {
                check1 = true;
                if (isValidHex(jTextField3.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        } else if (jComboBox8.getSelectedIndex() == 1) {
            if (jTextField3.getText().length() == 64) {
                check1 = true;
                if (isValidBinary(jTextField3.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        }
        if (jComboBox9.getSelectedIndex() == 0) {
            if (jTextField9.getText().length() == 8) {
                check2 = true;
            }
        } else if (jComboBox9.getSelectedIndex() == 1) {
            if (jTextField9.getText().length() == 16) {
                check2 = true;
                if (isValidHex(jTextField9.getText())) {
                    check2 = true;
                } else {
                    check2 = false;
                }
            }

        } else if (jComboBox9.getSelectedIndex() == 1) {
            if (jTextField9.getText().length() == 64) {
                check2 = true;
                if (isValidBinary(jTextField9.getText())) {
                    check2 = true;
                } else {
                    check2 = false;
                }
            }

        }

        if (check && check1 && check2) {
            if (jComboBox1.getSelectedIndex() == 0) {
                if ((jTextField1.getText().length() % 8 == 0)) {
                    if (jTextField1.getText().length() == 8) {
                        String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                        String[] keyArray2 =
```

```
keyArray(Efactore(jTextField3.getText(), 2));
                        String[] keyArray3 =
keyArray(Efactore(jTextField9.getText(), 4));
                        String binaryNumber0 =
Efactore(jTextField1.getText(), 0);
                        String binaryNumber1 = hexToBin(des(binaryNumber0,
keyArray1));
                        String binaryNumber2 = hexToBin(des(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des(binaryNumber2, keyArray3);
                        jTextField4.setText(Efactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < Efactore(jTextField1.getText(),
0).length(); i += groupSize) {
                            int endIndex = Math.min(i + groupSize,
Efactore(jTextField1.getText(), 0).length());
                            String group = Efactore(jTextField1.getText(),
0).substring(i, endIndex);
                            String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                            String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                            String[] keyArray3 =
keyArray(Efactore(jTextField9.getText(), 4));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des(binaryNumber1, keyArray2));
                            String binaryNumber3 = des(binaryNumber2,
keyArray3);
                            binaryNumber4 += binaryNumber3;

                        }

                        jTextField4.setText(Efactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 8, CHECK PLAIN TEXT");
                }

            } else if (jComboBox1.getSelectedIndex() == 1) {
                if ((jTextField1.getText().length() % 16 == 0)) {
                    if (jTextField1.getText().length() == 16) {
                        String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                        String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                        String[] keyArray3 =
keyArray(Efactore(jTextField9.getText(), 4));
                        String binaryNumber0 =
Efactore(jTextField1.getText(), 0);
                        String binaryNumber1 = hexToBin(des(binaryNumber0,
keyArray1));
```

```
                        String binaryNumber2 = hexToBin(des(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des(binaryNumber2, keyArray3);

                        jTextField4.setText(Efactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < Efactore(jTextField1.getText(),
0).length(); i += groupSize) {
                            int endIndex = Math.min(i + groupSize,
Efactore(jTextField1.getText(), 0).length());
                            String group = Efactore(jTextField1.getText(),
0).substring(i, endIndex);
                            String[] keyArray1 =
keyArray(Efactore(jTextField2.getText(), 1));
                            String[] keyArray2 =
keyArray(Efactore(jTextField3.getText(), 2));
                            String[] keyArray3 =
keyArray(Efactore(jTextField9.getText(), 4));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des(binaryNumber0, keyArray1));
                            String binaryNumber2 =
hexToBin(des(binaryNumber1, keyArray2));
                            String binaryNumber3 = des(binaryNumber2,
keyArray3);
                            binaryNumber4 += binaryNumber3;

                        }
                        jTextField4.setText(Efactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 16, CHECK PLAIN TEXT");
                }

            } else if (jComboBox1.getSelectedIndex() == 2) {
                if ((jTextField1.getText().length() % 64 == 0)) {
                    if (jTextField1.getText().length() == 64) {
                        String[] keyArray1 = keyArray(jTextField2.getText());
                        String[] keyArray2 = keyArray(jTextField3.getText());
                        String[] keyArray3 =
keyArray(Efactore(jTextField9.getText(), 4));
                        String binaryNumber0 =
Efactore(jTextField1.getText(), 0);
                        String binaryNumber1 = hexToBin(des(binaryNumber0,
keyArray1));
                        String binaryNumber2 = hexToBin(des(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des(binaryNumber2, keyArray3);

                        jTextField4.setText(Efactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < jTextField1.getText().length(); i
```

```
+= groupSize) {
                                int endIndex = Math.min(i + groupSize,
jTextField1.getText().length());
                                String group = jTextField1.getText().substring(i,
endIndex);
                                String[] keyArray1 =
keyArray(jTextField2.getText());
                                String[] keyArray2 =
keyArray(jTextField3.getText());
                                String[] keyArray3 =
keyArray(Efactore(jTextField9.getText(), 4));
                                String binaryNumber0 = group;
                                String binaryNumber1 =
hexToBin(des(binaryNumber0, keyArray1));
                                String binaryNumber2 =
hexToBin(des(binaryNumber1, keyArray2));
                                String binaryNumber3 = des(binaryNumber2,
keyArray3);
                                binaryNumber4 += binaryNumber3;

                            }
                            jTextField4.setText(Efactore(binaryNumber4, 3));
                        }
                    } else {
                        JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 64, CHECK PLAIN TEXT");
                    }
                }

            } else {
                JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF THE
INPUTS OF THE KEYS AND THE INPUT OF THE KEY FOR THE CHOSEN REPRESENTATION");
            }
        }//GEN-LAST:event_jButton1ActionPerformed

        private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton3ActionPerformed
            boolean check = false;
            boolean check1 = false;
            boolean check2 = false;
            if (jComboBox5.getSelectedIndex() == 0) {
                if (jTextField6.getText().length() == 8) {
                    check = true;
                }
            } else if (jComboBox5.getSelectedIndex() == 1) {
                if (jTextField6.getText().length() == 16) {
                    check = true;
                    if (isValidHex(jTextField6.getText())) {
                        check = true;
                    } else {
                        check = false;
                    }
                }

            } else if (jComboBox5.getSelectedIndex() == 1) {
                if (jTextField6.getText().length() == 64) {
                    check = true;
```

```
                if (isValidBinary(jTextField6.getText())) {
                    check = true;
                } else {
                    check = false;
                }
            }

        }

        if (jComboBox6.getSelectedIndex() == 0) {
            if (jTextField7.getText().length() == 8) {
                check1 = true;
            }
        } else if (jComboBox6.getSelectedIndex() == 1) {
            if (jTextField7.getText().length() == 16) {
                check1 = true;
                if (isValidHex(jTextField7.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        } else if (jComboBox6.getSelectedIndex() == 1) {
            if (jTextField7.getText().length() == 64) {
                check1 = true;
                if (isValidBinary(jTextField7.getText())) {
                    check1 = true;
                } else {
                    check1 = false;
                }
            }

        }

        if (jComboBox10.getSelectedIndex() == 0) {
            if (jTextField10.getText().length() == 8) {
                check2 = true;
            }
        } else if (jComboBox10.getSelectedIndex() == 1) {
            if (jTextField10.getText().length() == 16) {
                check2 = true;
                if (isValidHex(jTextField10.getText())) {
                    check2 = true;
                } else {
                    check2 = false;
                }
            }

        } else if (jComboBox10.getSelectedIndex() == 1) {
            if (jTextField10.getText().length() == 64) {
                check2 = true;
                if (isValidBinary(jTextField10.getText())) {
                    check2 = true;
                } else {
                    check2 = false;
                }
```

```
            }

        }
        if (check && check1 && check2) {

            if (jComboBox2.getSelectedIndex() == 0) {
                if ((jTextField5.getText().length() % 16 == 0)) {
                    if (jTextField5.getText().length() == 16) {
                        String[] keyArray1 =
keyArray(Dfactore(jTextField6.getText(), 1));
                        String[] keyArray2 =
keyArray(Dfactore(jTextField7.getText(), 2));
                        String[] keyArray3 =
keyArray(Dfactore(jTextField10.getText(), 4));
                        String binaryNumber0 =
Dfactore(jTextField5.getText(), 0);
                        String binaryNumber1 = hexToBin(des1(binaryNumber0,
keyArray3));
                        String binaryNumber2 = hexToBin(des1(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                        jTextField8.setText(Dfactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < Dfactore(jTextField5.getText(),
0).length(); i += groupSize) {
                            int endIndex = Math.min(i + groupSize,
Dfactore(jTextField5.getText(), 0).length());
                            String group = Dfactore(jTextField5.getText(),
0).substring(i, endIndex);
                            String[] keyArray1 =
keyArray(Dfactore(jTextField6.getText(), 1));
                            String[] keyArray2 =
keyArray(Dfactore(jTextField7.getText(), 2));
                            String[] keyArray3 =
keyArray(Dfactore(jTextField10.getText(), 4));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des1(binaryNumber0, keyArray3));
                            String binaryNumber2 =
hexToBin(des1(binaryNumber1, keyArray2));
                            String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;

                        }
                        jTextField8.setText(Dfactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 16, CHECK PLAIN TEXT");
                }

            } else if (jComboBox2.getSelectedIndex() == 1) {
                if ((jTextField5.getText().length() % 64 == 0)) {
```

```
                    if (jTextField5.getText().length() == 64) {
                        String[] keyArray1 =
keyArray(Dfactore(jTextField6.getText(), 1));
                        String[] keyArray2 =
keyArray(Dfactore(jTextField7.getText(), 2));
                        String[] keyArray3 =
keyArray(Dfactore(jTextField10.getText(), 4));
                        String binaryNumber0 =
Dfactore(jTextField5.getText(), 0);
                        String binaryNumber1 = hexToBin(des1(binaryNumber0,
keyArray3));
                        String binaryNumber2 = hexToBin(des1(binaryNumber1,
keyArray2));
                        String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                        jTextField8.setText(Dfactore(binaryNumber3, 3));
                    } else {
                        String binaryNumber4 = "";
                        int groupSize = 64;
                        for (int i = 0; i < jTextField5.getText().length(); i
+= groupSize) {
                            int endIndex = Math.min(i + groupSize,
jTextField5.getText().length());
                            String group = jTextField5.getText().substring(i,
endIndex);
                            String[] keyArray1 =
keyArray(jTextField6.getText());
                            String[] keyArray2 =
keyArray(jTextField7.getText());
                            String[] keyArray3 =
keyArray(Dfactore(jTextField10.getText(), 4));
                            String binaryNumber0 = group;
                            String binaryNumber1 =
hexToBin(des1(binaryNumber0, keyArray3));
                            String binaryNumber2 =
hexToBin(des1(binaryNumber1, keyArray2));
                            String binaryNumber3 = des1(binaryNumber2,
keyArray1);
                            binaryNumber4 += binaryNumber3;

                        }
                        jTextField8.setText(Dfactore(binaryNumber4, 3));
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF
THE INPUTS IS MULTIPLE OF 64, CHECK PLAIN TEXT");
                }
            }

        } else {
            JOptionPane.showMessageDialog(null, "CHECK THE LENGTH OF THE
INPUTS OF THE KEYS AND THE INPUT OF THE KEY FOR THE CHOSEN REPRESENTATION");
        }

    }//GEN-LAST:event_jButton3ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
```

```
{//GEN-FIRST:event_jButton4ActionPerformed
        menu menu = new menu();
        menu.show();
        dispose();
    }//GEN-LAST:event_jButton4ActionPerformed

    private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jComboBox1ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_jComboBox1ActionPerformed
    public static String textToBinary(String text) {
        byte[] bytes = text.getBytes();
        StringBuilder binaryStringBuilder = new StringBuilder();
        for (byte b : bytes) {
            int value = b;
            for (int i = 7; i >= 0; i--) {
                binaryStringBuilder.append((value & (1 << i)) == 0 ? '0' :
'1');
            }
            // Add a space between bytes for readability
        }
        return binaryStringBuilder.toString();
    }

    public static String hexToText(String hex) {
        byte[] bytes = new byte[hex.length() / 2];
        for (int i = 0; i < hex.length(); i += 2) {
            int byteValue = Integer.parseInt(hex.substring(i, i + 2), 16);
            bytes[i / 2] = (byte) byteValue;
        }

        try {
            return new String(bytes, "UTF-8");
        } catch (UnsupportedEncodingException ex) {
            Logger.getLogger(Three.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return null;

    }

    public static String des(String binaryString, String[] keys) {
        System.out.println("first " + binaryString);
        // Define the initial permutation table for DES
        int[] initialPermutationTable = {
            58, 50, 42, 34, 26, 18, 10, 2,
            60, 52, 44, 36, 28, 20, 12, 4,
            62, 54, 46, 38, 30, 22, 14, 6,
            64, 56, 48, 40, 32, 24, 16, 8,
            57, 49, 41, 33, 25, 17, 9, 1,
            59, 51, 43, 35, 27, 19, 11, 3,
            61, 53, 45, 37, 29, 21, 13, 5,
            63, 55, 47, 39, 31, 23, 15, 7
        };

        // Perform the initial permutation
        StringBuilder ipresult = new StringBuilder();
```

```
        for (int i : initialPermutationTable) {
            ipresult.append(binaryString.charAt(i - 1));
        }
        System.out.println("after ip " + ipresult);
        String[][] ipArrayResult = new String[17][2];
        StringBuilder sbresultl = new StringBuilder();
        StringBuilder sbresultr = new StringBuilder();
        for (int j = 0; j < 64; j++) {

            if (j < 32) {
                sbresultl.append(ipresult.charAt(j));
            }
            if (j >= 32) {
                sbresultr.append(ipresult.charAt(j));
            }
            ipArrayResult[0][0] = sbresultl.toString();
            ipArrayResult[0][1] = sbresultr.toString();
        }
        System.out.println("l0  " + ipArrayResult[0][0]);
        System.out.println("r0 " + ipArrayResult[0][1]);
        for (int i = 1; i < 17; i++) {

            ipArrayResult[i][0] = ipArrayResult[i - 1][1];
            System.out.println("l" + i + " " + ipArrayResult[i][0]);
            int[] eBitSelectionTable = {
                32, 1, 2, 3, 4, 5,
                4, 5, 6, 7, 8, 9,
                8, 9, 10, 11, 12, 13,
                12, 13, 14, 15, 16, 17,
                16, 17, 18, 19, 20, 21,
                20, 21, 22, 23, 24, 25,
                24, 25, 26, 27, 28, 29,
                28, 29, 30, 31, 32, 1
            };
            StringBuilder eresult = new StringBuilder();
            for (int j : eBitSelectionTable) {
                eresult.append(ipArrayResult[i - 1][1].charAt(j - 1));
            }
            //0110111111
            System.out.println("rebit " + eresult);
            System.out.println("key" + (i - 1) + " " + keys[i - 1]);
            StringBuilder xourEKresult = new StringBuilder();
            for (int j = 0; j < eresult.length(); j++) {
                xourEKresult.append((char) ('0' + (eresult.charAt(j) ^ keys[i
- 1].charAt(j))));
            }
            System.out.println("xorEbitKey " + xourEKresult);

            String[] sboxArray = new String[8];
            int count = 0;
            StringBuilder ssresult = new StringBuilder();
            for (int j = 0; j < 48; j++) {
                if (j % 6 == 0 && j > 0) {
                    sboxArray[count++] = ssresult.toString();
                    ssresult = new StringBuilder();
                }
                ssresult.append(xourEKresult.charAt(j));
```

```
            }
            if (ssresult.length() > 0) {
                sboxArray[count] = ssresult.toString();
            }

            StringBuilder sboxsresult = new StringBuilder();
            int[][] s1Box = {
                {14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7},
                {0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8},
                {4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0},
                {15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13}
            };
            String x = String.valueOf(sboxArray[0].charAt(0) + "" +
sboxArray[0].charAt(sboxArray[0].length() - 1));
            String y = String.valueOf(sboxArray[0].substring(1, 5));
            int row = Integer.parseInt(x, 2);
            int colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s1Box[row][colm])).replace(' ', '0'));

            int[][] s2Box = {
                {15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10},
                {3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5},
                {0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15},
                {13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9}
            };
            x = String.valueOf(sboxArray[1].charAt(0) + "" +
sboxArray[1].charAt(sboxArray[1].length() - 1));
            y = String.valueOf(sboxArray[1].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s2Box[row][colm])).replace(' ', '0'));
            int[][] s3Box = {
                {10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8},
                {13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1},
                {13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7},
                {1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12}
            };
            x = String.valueOf(sboxArray[2].charAt(0) + "" +
sboxArray[2].charAt(sboxArray[2].length() - 1));
            y = String.valueOf(sboxArray[2].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s3Box[row][colm])).replace(' ', '0'));
            if (i == 6) {

                System.out.println(sboxsresult.charAt(6));
            }
            int[][] s4Box = {
                {7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15},
                {13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9},
                {10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4},
                {3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14}
            };
            x = String.valueOf(sboxArray[3].charAt(0) + "" +
```

```
sboxArray[3].charAt(sboxArray[3].length() - 1));
            y = String.valueOf(sboxArray[3].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s4Box[row][colm])).replace(' ', '0'));
            int[][] s5Box = {
                {2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9},
                {14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6},
                {4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14},
                {11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3}
            };
            x = String.valueOf(sboxArray[4].charAt(0) + "" +
sboxArray[4].charAt(sboxArray[4].length() - 1));
            y = String.valueOf(sboxArray[4].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s5Box[row][colm])).replace(' ', '0'));
            int[][] s6Box = {
                {12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11},
                {10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8},
                {9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6},
                {4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13}
            };
            x = String.valueOf(sboxArray[5].charAt(0) + "" +
sboxArray[5].charAt(sboxArray[5].length() - 1));
            y = String.valueOf(sboxArray[5].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s6Box[row][colm])).replace(' ', '0'));

            int[][] s7Box = {
                {4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1},
                {13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6},
                {1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2},
                {6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12}
            };
            x = String.valueOf(sboxArray[6].charAt(0) + "" +
sboxArray[6].charAt(sboxArray[6].length() - 1));
            y = String.valueOf(sboxArray[6].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s7Box[row][colm])).replace(' ', '0'));

            int[][] s8Box = {
                {13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7},
                {1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2},
                {7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8},
                {2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11}
            };
            x = String.valueOf(sboxArray[7].charAt(0) + "" +
sboxArray[7].charAt(sboxArray[7].length() - 1));
            y = String.valueOf(sboxArray[7].substring(1, 5));
            row = Integer.parseInt(x, 2);
```

```
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s8Box[row][colm])).replace(' ', '0'));
            System.out.println("sbox " + sboxsresult);

            int[] pPermutationTable = {
                16, 7, 20, 21,
                29, 12, 28, 17,
                1, 15, 23, 26,
                5, 18, 31, 10,
                2, 8, 24, 14,
                32, 27, 3, 9,
                19, 13, 30, 6,
                22, 11, 4, 25
            };

            StringBuilder pPermutat = new StringBuilder();
            for (int j : pPermutationTable) {
                pPermutat.append(sboxsresult.charAt(j - 1));
            }
            System.out.println("pbox " + pPermutat);
            System.out.println("left " + (i - 1) + " " + ipArrayResult[i -
1][0]);
            StringBuilder xourplresult = new StringBuilder();
            for (int j = 0; j < pPermutat.length(); j++) {
                xourplresult.append((char) ('0' + (pPermutat.charAt(j) ^
ipArrayResult[i - 1][0].charAt(j))));
            }
            System.out.println("pbox " + xourplresult);
            ipArrayResult[i][1] = xourplresult.toString();
            System.out.println("r " + i + " " + xourplresult);
            // keyArray[i]=eresult.toString();
            if (i == 2) {
                System.out.println();

            }
        }
        System.out.println("end");
        String x = ipArrayResult[16][0];
        System.out.println("l " + x);
        String y = ipArrayResult[16][1];
        System.out.println("r " + y);
        String z = y + "" + x;
        System.out.println("beforinv " + z);
        int[] invip1PermutationTable = {
            40, 8, 48, 16, 56, 24, 64, 32,
            39, 7, 47, 15, 55, 23, 63, 31,
            38, 6, 46, 14, 54, 22, 62, 30,
            37, 5, 45, 13, 53, 21, 61, 29,
            36, 4, 44, 12, 52, 20, 60, 28,
            35, 3, 43, 11, 51, 19, 59, 27,
            34, 2, 42, 10, 50, 18, 58, 26,
            33, 1, 41, 9, 49, 17, 57, 25
        };
        StringBuilder invPermutat = new StringBuilder();
        for (int j : invip1PermutationTable) {
            invPermutat.append(z.charAt(j - 1));
```

```
        }
        System.out.println("afterinv " + invPermutat);

        BigInteger decimal = new BigInteger(String.valueOf(invPermutat), 2);

        // Convert decimal to hexadecimal
        String hexString = decimal.toString(16);
        System.out.println("to hex " + hexString);
        return hexString;
    }

    public static String des1(String binaryString, String[] keys) {
        System.out.println("first " + binaryString);
        // Define the initial permutation table for DES
        int[] initialPermutationTable = {
            58, 50, 42, 34, 26, 18, 10, 2,
            60, 52, 44, 36, 28, 20, 12, 4,
            62, 54, 46, 38, 30, 22, 14, 6,
            64, 56, 48, 40, 32, 24, 16, 8,
            57, 49, 41, 33, 25, 17, 9, 1,
            59, 51, 43, 35, 27, 19, 11, 3,
            61, 53, 45, 37, 29, 21, 13, 5,
            63, 55, 47, 39, 31, 23, 15, 7
        };

        // Perform the initial permutation
        StringBuilder ipresult = new StringBuilder();
        for (int i : initialPermutationTable) {
            ipresult.append(binaryString.charAt(i - 1));
        }
        System.out.println("after ip " + ipresult);
        String[][] ipArrayResult = new String[17][2];
        StringBuilder sbresultl = new StringBuilder();
        StringBuilder sbresultr = new StringBuilder();
        for (int j = 0; j < 64; j++) {

            if (j < 32) {
                sbresultl.append(ipresult.charAt(j));
            }
            if (j >= 32) {
                sbresultr.append(ipresult.charAt(j));
            }
            ipArrayResult[0][0] = sbresultl.toString();
            ipArrayResult[0][1] = sbresultr.toString();
        }
        System.out.println("l0  " + ipArrayResult[0][0]);
        System.out.println("r0 " + ipArrayResult[0][1]);
        for (int i = 1; i < 17; i++) {

            ipArrayResult[i][0] = ipArrayResult[i - 1][1];
            System.out.println("l" + i + " " + ipArrayResult[i][0]);
            int[] eBitSelectionTable = {
                32, 1, 2, 3, 4, 5,
                4, 5, 6, 7, 8, 9,
                8, 9, 10, 11, 12, 13,
                12, 13, 14, 15, 16, 17,
                16, 17, 18, 19, 20, 21,
```

```
                20, 21, 22, 23, 24, 25,
                24, 25, 26, 27, 28, 29,
                28, 29, 30, 31, 32, 1
            };
            StringBuilder eresult = new StringBuilder();
            for (int j : eBitSelectionTable) {
                eresult.append(ipArrayResult[i - 1][1].charAt(j - 1));
            }
            //0110111111
            System.out.println("rebit " + eresult);
            System.out.println("key" + (i - 1) + " " + keys[(16 - i)]);
            StringBuilder xourEKresult = new StringBuilder();
            for (int j = 0; j < eresult.length(); j++) {
                xourEKresult.append((char) ('0' + (eresult.charAt(j) ^
keys[(16 - i)].charAt(j))));
            }
            System.out.println("xorEbitKey " + xourEKresult);

            String[] sboxArray = new String[8];
            int count = 0;
            StringBuilder ssresult = new StringBuilder();
            for (int j = 0; j < 48; j++) {
                if (j % 6 == 0 && j > 0) {
                    sboxArray[count++] = ssresult.toString();
                    ssresult = new StringBuilder();
                }
                ssresult.append(xourEKresult.charAt(j));
            }
            if (ssresult.length() > 0) {
                sboxArray[count] = ssresult.toString();
            }

            StringBuilder sboxsresult = new StringBuilder();
            int[][] s1Box = {
                {14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7},
                {0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8},
                {4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0},
                {15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13}
            };
            String x = String.valueOf(sboxArray[0].charAt(0) + "" +
sboxArray[0].charAt(sboxArray[0].length() - 1));
            String y = String.valueOf(sboxArray[0].substring(1, 5));
            int row = Integer.parseInt(x, 2);
            int colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s1Box[row][colm])).replace(' ', '0'));

            int[][] s2Box = {
                {15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10},
                {3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5},
                {0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15},
                {13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9}
            };
            x = String.valueOf(sboxArray[1].charAt(0) + "" +
sboxArray[1].charAt(sboxArray[1].length() - 1));
            y = String.valueOf(sboxArray[1].substring(1, 5));
            row = Integer.parseInt(x, 2);
```

```
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s2Box[row][colm])).replace(' ', '0'));
            int[][] s3Box = {
                {10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8},
                {13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1},
                {13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7},
                {1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12}
            };
            x = String.valueOf(sboxArray[2].charAt(0) + "" +
sboxArray[2].charAt(sboxArray[2].length() - 1));
            y = String.valueOf(sboxArray[2].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s3Box[row][colm])).replace(' ', '0'));
            if (i == 6) {

                System.out.println(sboxsresult.charAt(6));
            }
            int[][] s4Box = {
                {7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15},
                {13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9},
                {10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4},
                {3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14}
            };
            x = String.valueOf(sboxArray[3].charAt(0) + "" +
sboxArray[3].charAt(sboxArray[3].length() - 1));
            y = String.valueOf(sboxArray[3].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s4Box[row][colm])).replace(' ', '0'));
            int[][] s5Box = {
                {2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9},
                {14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6},
                {4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14},
                {11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3}
            };
            x = String.valueOf(sboxArray[4].charAt(0) + "" +
sboxArray[4].charAt(sboxArray[4].length() - 1));
            y = String.valueOf(sboxArray[4].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s5Box[row][colm])).replace(' ', '0'));
            int[][] s6Box = {
                {12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11},
                {10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8},
                {9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6},
                {4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13}
            };
            x = String.valueOf(sboxArray[5].charAt(0) + "" +
sboxArray[5].charAt(sboxArray[5].length() - 1));
            y = String.valueOf(sboxArray[5].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
```

```
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s6Box[row][colm])).replace(' ', '0'));

            int[][] s7Box = {
                {4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1},
                {13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6},
                {1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2},
                {6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12}
            };
            x = String.valueOf(sboxArray[6].charAt(0) + "" +
sboxArray[6].charAt(sboxArray[6].length() - 1));
            y = String.valueOf(sboxArray[6].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s7Box[row][colm])).replace(' ', '0'));

            int[][] s8Box = {
                {13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7},
                {1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2},
                {7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8},
                {2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11}
            };
            x = String.valueOf(sboxArray[7].charAt(0) + "" +
sboxArray[7].charAt(sboxArray[7].length() - 1));
            y = String.valueOf(sboxArray[7].substring(1, 5));
            row = Integer.parseInt(x, 2);
            colm = Integer.parseInt(y, 2);
            sboxsresult.append(String.format("%4s",
Integer.toBinaryString(s8Box[row][colm])).replace(' ', '0'));
            System.out.println("sbox " + sboxsresult);

            int[] pPermutationTable = {
                16, 7, 20, 21,
                29, 12, 28, 17,
                1, 15, 23, 26,
                5, 18, 31, 10,
                2, 8, 24, 14,
                32, 27, 3, 9,
                19, 13, 30, 6,
                22, 11, 4, 25
            };

            StringBuilder pPermutat = new StringBuilder();
            for (int j : pPermutationTable) {
                pPermutat.append(sboxsresult.charAt(j - 1));
            }
            System.out.println("pbox " + pPermutat);
            System.out.println("left " + (i - 1) + " " + ipArrayResult[i -
1][0]);
            StringBuilder xourplresult = new StringBuilder();
            for (int j = 0; j < pPermutat.length(); j++) {
                xourplresult.append((char) ('0' + (pPermutat.charAt(j) ^
ipArrayResult[i - 1][0].charAt(j))));
            }
            System.out.println("pbox " + xourplresult);
            ipArrayResult[i][1] = xourplresult.toString();
```

```
            System.out.println("r " + i + " " + xourplresult);
            // keyArray[i]=eresult.toString();
            if (i == 2) {
                System.out.println();

            }
        }
        System.out.println("end");
        String x = ipArrayResult[16][0];
        System.out.println("l " + x);
        String y = ipArrayResult[16][1];
        System.out.println("r " + y);
        String z = y + "" + x;
        System.out.println("beforinv " + z);
        int[] invip1PermutationTable = {
            40, 8, 48, 16, 56, 24, 64, 32,
            39, 7, 47, 15, 55, 23, 63, 31,
            38, 6, 46, 14, 54, 22, 62, 30,
            37, 5, 45, 13, 53, 21, 61, 29,
            36, 4, 44, 12, 52, 20, 60, 28,
            35, 3, 43, 11, 51, 19, 59, 27,
            34, 2, 42, 10, 50, 18, 58, 26,
            33, 1, 41, 9, 49, 17, 57, 25
        };
        StringBuilder invPermutat = new StringBuilder();
        for (int j : invip1PermutationTable) {
            invPermutat.append(z.charAt(j - 1));
        }
        System.out.println("afterinv " + invPermutat);

        BigInteger decimal = new BigInteger(String.valueOf(invPermutat), 2);

        // Convert decimal to hexadecimal
        String hexString = decimal.toString(16);
        System.out.println("to hex " + hexString);
        return hexString;
    }

    public static String hexToBin(String hex) {
        StringBuilder binary = new StringBuilder();
        for (int i = 0; i < hex.length(); i++) {
            char hexDigit = hex.charAt(i);
            int decimalValue = Character.digit(hexDigit, 16);
            String binaryValue = Integer.toBinaryString(decimalValue);
            // Ensure each binary representation has 4 bits
            while (binaryValue.length() < 4) {
                binaryValue = "0" + binaryValue;
            }
            binary.append(binaryValue);
        }
        return binary.toString();
    }

    public static String binToHex(String binary) {
        StringBuilder hex = new StringBuilder();
        for (int i = 0; i < binary.length(); i += 4) {
            // Ensure that the substring has 4 bits, otherwise add zeros to
```

```
the left
            String binSub = binary.substring(i, Math.min(i + 4,
binary.length()));
            while (binSub.length() < 4) {
                binSub = "0" + binSub;
            }

            int decimalValue = Integer.parseInt(binSub, 2);
            String hexValue =
Integer.toHexString(decimalValue).toUpperCase();

            hex.append(hexValue);
        }
        return hex.toString();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Three.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Three.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Three.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Three.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        System.out.println(binToHex(textToBinary("bbbbbbbb")));
        System.out.println(binToHex(textToBinary("cccccccc")));
```

```
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Three().setVisible(true);
            }
        });
    }

    public static String[] keyArray(String theKey) {

        // Define the initial permutation table for DES
        int[] pc1Table = {
            57, 49, 41, 33, 25, 17, 9,
            1, 58, 50, 42, 34, 26, 18,
            10, 2, 59, 51, 43, 35, 27,
            19, 11, 3, 60, 52, 44, 36,
            63, 55, 47, 39, 31, 23, 15,
            7, 62, 54, 46, 38, 30, 22,
            14, 6, 61, 53, 45, 37, 29,
            21, 13, 5, 28, 20, 12, 4
        };

        // Perform the initial permutation
        StringBuilder pc1result = new StringBuilder();
        for (int i : pc1Table) {
            pc1result.append(theKey.charAt(i - 1));
        }
        String binaryString = pc1result.toString();

        if (binaryString.length() % 2 != 0) {
            throw new IllegalArgumentException("Binary string length must be
even.");
        }
        String[][] pc1ArrayResult = new String[17][2];
        StringBuilder sbresultl = new StringBuilder();
        StringBuilder sbresultr = new StringBuilder();
        for (int j = 0; j < 56; j++) {

            if (j < 28) {
                sbresultl.append(binaryString.charAt(j));
            }
            if (j >= 28) {
                sbresultr.append(binaryString.charAt(j));
            }
            pc1ArrayResult[0][0] = sbresultl.toString();
            pc1ArrayResult[0][1] = sbresultr.toString();
        }

        for (int i = 1; i < 17; i++) {
            if (i == 1 || i == 2 || i == 9 || i == 16) {
                pc1ArrayResult[i][0] = shiftLeft(pc1ArrayResult[i - 1][0],
1);
                pc1ArrayResult[i][1] = shiftLeft(pc1ArrayResult[i - 1][1],
1);
            } else {

                pc1ArrayResult[i][0] = shiftLeft(pc1ArrayResult[i - 1][0],
```

```
2);
                pc1ArrayResult[i][1] = shiftLeft(pc1ArrayResult[i - 1][1],
2);
            }
        }
        String[] keyArray = new String[16];

        int[] pc2Table = {
            14, 17, 11, 24, 1, 5,
            3, 28, 15, 6, 21, 10,
            23, 19, 12, 4, 26, 8,
            16, 7, 27, 20, 13, 2,
            41, 52, 31, 37, 47, 55,
            30, 40, 51, 45, 33, 48,
            44, 49, 39, 56, 34, 53,
            46, 42, 50, 36, 29, 32
        };

        for (int i = 0; i < 16; i++) {
            keyArray[i] = pc1ArrayResult[i + 1][0] + pc1ArrayResult[i +
1][1];
            StringBuilder pc2result = new StringBuilder();
            for (int j : pc2Table) {
                pc2result.append(keyArray[i].charAt(j - 1));
            }
            keyArray[i] = pc2result.toString();
        }
        return keyArray;
    }

    public static String shiftLeft(String binaryString, int positions) {
        // Convert the binary string to a char array for easy manipulation
        char[] charArray = binaryString.toCharArray();

        // Perform the left shift for the specified number of positions
        for (int shift = 0; shift < positions; shift++) {
            char firstChar = charArray[0];

            for (int i = 1; i < charArray.length; i++) {
                charArray[i - 1] = charArray[i];
            }

            charArray[charArray.length - 1] = firstChar;
        }

        // Convert the char array back to a string
        return new String(charArray);
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox10;
    private javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JComboBox<String> jComboBox3;
    private javax.swing.JComboBox<String> jComboBox4;
```

```
    private javax.swing.JComboBox<String> jComboBox5;
    private javax.swing.JComboBox<String> jComboBox6;
    private javax.swing.JComboBox<String> jComboBox7;
    private javax.swing.JComboBox<String> jComboBox8;
    private javax.swing.JComboBox<String> jComboBox9;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField10;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    // End of variables declaration//GEN-END:variables
}
```