import javax.swing.\*;  
import java.awt.\*;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
  
public class Stego {  
  
 public static void operate (int key){  
 JFileChooser fileChooser = new JFileChooser(); // to choose file  
 fileChooser.showOpenDialog(null);  
 File file = fileChooser.getSelectedFile();  
  
 try {  
 FileInputStream fileInputStream = new FileInputStream(file);  
 byte []data = new byte[fileInputStream.available()]; //creating an array  
 fileInputStream.read(data); //image taken will be converted on byte  
 int i=0;  
  
 for (byte b: data){  
 System.*out*.println(b);  
 data[i] = (byte) (b^key); // for performing encryption and decryption, XOR function is used.  
 i++;  
 }  
 FileOutputStream fileOutputStream = new FileOutputStream(file);  
 fileOutputStream.write(data);  
 fileOutputStream.close();  
 fileInputStream.close();  
 JOptionPane.*showMessageDialog*(null, "Done");  
  
 }catch (Exception e){  
 e.printStackTrace();  
 }  
 }  
 public static void main(String args[]){  
  
 System.*out*.println("Welcome!! Enter the key and choose an image.");  
  
 JFrame f = new JFrame();  
 f.setTitle("Encrypt and Decrypt"); // sets title  
 f.setSize(500,250); //sets size  
 f.setLocationRelativeTo(null); // aligned to center  
 f.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*); // for effective exiting when app is closed  
  
 Font font = new Font("Roboto",Font.*BOLD*, 20);  
  
 // creating button  
 JButton button = new JButton();  
 button.setText("Select any image"); // sets text  
 button.setFont(font);// sets font  
 button.setBackground(Color.*yellow*);  
  
 // creating text field  
 JTextField textField = new JTextField(25);  
 textField.setBackground(Color.*green*);  
 textField.setFont(font);  
  
 // creating labels  
 JLabel label1 = new JLabel();  
 label1.setText("Enter the secret key");  
 label1.setFont(font);  
  
 JLabel label2 = new JLabel();  
 label2.setText(" Select image for encryption/decryption ");  
 label2.setFont(font);  
  
 button.addActionListener(e -> { // ActionListener is an interface  
 System.*out*.println("Button clicked");  
 String text= textField.getText(); //takes text input  
 long temp = Long.*parseLong*(text); // type casting  
 *operate*((int) temp);  
 });  
  
 f.setLayout(new FlowLayout()); //used to arrange components in a flow  
 f.add(label1);  
 f.add(textField);  
 f.add(label2);  
 f.add(button);  
 f.setVisible(true);  
 }  
}