//Main.java

package com.example.test2;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class Main extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(Main.class.getResource("Scene1.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 700, 600);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public static void main(String[] args) {  
 launch();  
 }  
}

//Scene1.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.Scene1Controller">  
 <Label fx:id="questionText" layoutX="44.0" layoutY="43.0" prefHeight="60.0" prefWidth="527.0" text="What would you like to do?">  
 <font>  
 <Font name="Andale Mono" size="33.0" />  
 </font>  
 </Label>  
 <Button fx:id="decrypt" layoutX="367.0" layoutY="183.0" mnemonicParsing="false" onAction="#D" text="Decrypt">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 <Button fx:id="encrypt" layoutX="133.0" layoutY="183.0" mnemonicParsing="false" onAction="#E" text="Encrypt">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
</AnchorPane>

//Scene1Controller

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.stage.Stage;  
import java.io.IOException;  
import java.util.Objects;  
  
public class Scene1Controller {  
 private Stage stage;  
 private Scene scene;  
 private Parent root;  
  
 @FXML  
 private Label questionText;  
  
 @FXML  
 private Button encrypt;  
  
 @FXML  
 private Button decrypt;  
  
 public void E(ActionEvent event) throws IOException { //Allows users to press "Encrypt" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("EncryptionOptionsScreen.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
  
 public void D(ActionEvent event) throws IOException { //Allows users to press "Decrypt" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("DecryptionOptionsScreen.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
}

//Caesar.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.CaesarController">  
 <children>  
 <Label fx:id="questionCaesar" layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="400.0" text="Please Enter Plaintext" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="28.0" />  
 </font></Label>  
 <TextField fx:id="CaesarTextField" layoutX="116.0" layoutY="185.0">  
 <font>  
 <Font size="30.0" />  
 </font>  
 </TextField>  
 <Button fx:id="submit" layoutX="259.0" layoutY="262.0" mnemonicParsing="false" onAction="#submit" text="Submit">  
 <font>  
 <Font size="19.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

//CaesarController.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
  
  
  
  
public class CaesarController {  
  
 @FXML  
 private Label questionCaesar;  
  
 @FXML  
 private Button submit;  
  
 @FXML  
 private TextField CaesarTextField;  
  
 String input;  
  
 public void submit(ActionEvent event){  
 input = CaesarTextField.getText(); // Taking the user input and storing it into "input" variable  
 char[] inputChars = input.toCharArray(); // Creating an array named "char" and storing the individuals characters  
  
 String str ="";  
  
 for(char i: inputChars){ // incrementing the position of the characters according to ASCII or shifting right by 3  
 i+=3; // number of keys in this case 3  
 str = str + i;  
 System.out.print(i);  
  
  
 }  
 questionCaesar.setText("Output: " + str);  
  
  
  
  
  
  
  
 }  
  
  
}

//CaesarControllerD.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
  
  
// Handles the button functions for the Caesar Decryption  
public class CaesarControllerD {  
  
 @FXML  
 private Label questionCaesarD;  
  
 @FXML  
 private Button submitD;  
  
 @FXML  
 private TextField CaesarTextFieldD;  
  
 String inputD;  
  
 public void submit(ActionEvent event){  
 //Getting the input from the textfield  
 inputD = CaesarTextFieldD.getText();  
  
 //Storing the input into a character array  
 char[] inputChars = inputD.toCharArray();  
  
 String str = "";  
  
 // Subtracting the ASCii value by 3 to simulate the shift  
 for(char i: inputChars){  
 i-=3; // Shift of 3  
 str = str + i;  
 System.out.print(i);  
 }  
 questionCaesarD.setText("Output: " + str);  
  
  
  
  
  
  
  
 }  
  
  
}

//CaesarD.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.CaesarControllerD">  
 <children>  
 <Label fx:id="questionCaesarD" layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="381.0" text="Please Enter Ciphertext" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="27.0" />  
 </font></Label>  
 <TextField fx:id="CaesarTextFieldD" layoutX="116.0" layoutY="185.0">  
 <font>  
 <Font size="30.0" />  
 </font>  
 </TextField>  
 <Button fx:id="submitD" layoutX="259.0" layoutY="262.0" mnemonicParsing="false" onAction="#submit" text="Submit">  
 <font>  
 <Font size="19.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

//DecryptionOptionsScreen.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.DecryptionOptionsController">  
 <children>  
 <Button fx:id="caesaDr" layoutX="110.0" layoutY="188.0" mnemonicParsing="false" onAction="#switchtoCaesarD" text="Caesar">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 <Button fx:id="desD" layoutX="434.0" layoutY="188.0" mnemonicParsing="false" onAction="#switchtoDESD" text="DES">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 <Label layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="381.0" text="Decryption Options" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="33.0" />  
 </font></Label>  
 <Button fx:id="hillD" layoutX="276.0" layoutY="188.0" mnemonicParsing="false" onAction="#switchtoHillD" text="Hill">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

//DecryptionOptionsController.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
import java.util.Objects;  
  
public class DecryptionOptionsController {  
 private Stage stage;  
 private Scene scene;  
  
 @FXML  
 private Label questionText1;  
  
 @FXML  
 private Button caesaDr;  
  
 @FXML  
 private Button hillD;  
  
 @FXML  
 private Button desD;  
  
 public void switchtoCaesarD(ActionEvent event) throws IOException { //Allows users to press "Caesar" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("CaesarD.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
  
 public void switchtoHillD(ActionEvent event) throws IOException { //Allows users to press "Hill" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("HillD.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
 public void switchtoDESD(ActionEvent event) throws IOException { //Allows users to press "DES" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("DESD.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
}

//EncryptionOptionsScreen.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane fx:id="back" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.EncryptionOptionsController">  
 <children>  
 <Button fx:id="caesar" layoutX="110.0" layoutY="188.0" mnemonicParsing="false" onAction="#switchtoCaesar" text="Caesar">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 <Button fx:id="des" layoutX="434.0" layoutY="188.0" mnemonicParsing="false" onAction="#switchtoDES" text="DES">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 <Label layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="381.0" text="Encryption Options" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="33.0" />  
 </font></Label>  
 <Button fx:id="hill" layoutX="276.0" layoutY="188.0" mnemonicParsing="false" onAction="#switchtoHill" text="Hill">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

//EncryptionOptionsController.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
import java.util.Objects;  
  
public class EncryptionOptionsController {  
 private Stage stage;  
 private Scene scene;  
 private Parent root;  
 @FXML  
 private Label questionText1;  
  
 @FXML  
 private Button caesar;  
  
 @FXML  
 private Button hill;  
  
 @FXML  
 private Button des;  
  
 public void switchtoCaesar(ActionEvent event) throws IOException { //Allows users to press "Caesar" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Caesar.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
  
 public void switchtoHill(ActionEvent event) throws IOException { //Allows users to press "Hill" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Hill.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
 public void switchtoDES(ActionEvent event) throws IOException { //Allows users to press "DES" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("DES.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
 public void back(ActionEvent event) throws IOException { //Allows users to press "Back" button  
 Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Scene1.fxml")));  
 stage = (Stage)((Node)event.getSource()).getScene().getWindow();  
 scene = new Scene(root);  
 stage.setScene(scene);  
 stage.show();  
 }  
}

//DES.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.DESController">  
 <children>  
 <Label fx:id="questionDes" layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="400.0" text="Please Enter Plaintext" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="28.0" />  
 </font></Label>  
 <TextField fx:id="DesTextField" layoutX="116.0" layoutY="185.0">  
 <font>  
 <Font size="30.0" />  
 </font>  
 </TextField>  
 <Button fx:id="submitDes" layoutX="259.0" layoutY="262.0" mnemonicParsing="false" onAction="#submitDes" text="Submit">  
 <font>  
 <Font size="19.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

//DESController.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import java.security.InvalidKeyException;  
import java.security.NoSuchAlgorithmException;  
import javax.crypto.Cipher;  
import javax.crypto.IllegalBlockSizeException;  
import javax.crypto.KeyGenerator;  
import javax.crypto.SecretKey;  
import java.util.Arrays;  
import javax.crypto.\*;  
  
  
  
  
//https://mkyong.com/java/jce-encryption-data-encryption-standard-des-tutorial/  
//Parts of the code was derived from the website above.  
  
public class DESController {  
 //Declaring a variety of buttons, labels, text fields for the user to see  
 @FXML  
 private Label questionDes;  
  
 @FXML  
 private Button submitDes;  
  
 @FXML  
 private TextField DesTextField;  
  
 String inputDes;  
  
 public void submitDes(ActionEvent event){  
 try{  
 //This is a key generator instance imported from the DES library to generate a unique key.  
 KeyGenerator keyGenerator = KeyGenerator.getInstance("DES");  
 SecretKey myDesKey = keyGenerator.generateKey();  
 Cipher desCipher;  
  
 //Constructs the Cipher from the imported instance  
 desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");  
  
 //Initializing the Cipher for encryption function  
 desCipher.init(Cipher.ENCRYPT\_MODE, myDesKey);  
  
 //Storing the input gathered from the textfield into a string named inputDes  
 String inputDes = DesTextField.getText();  
  
 //Storing the input into a byte array after converting the input into bytes  
 byte[] bytesInput = inputDes.getBytes();  
  
 byte[] inputEncrypted = desCipher.doFinal(bytesInput);  
  
  
 //Gathering the output and changing the label to display the output of the encryption  
 questionDes.setText("Output: " + (inputEncrypted));  
  
  
  
 // A variety of exceptions included in the library  
 }catch(NoSuchAlgorithmException e){  
 e.printStackTrace();  
 }catch(NoSuchPaddingException e){  
 e.printStackTrace();  
 } catch(IllegalBlockSizeException e){  
 e.printStackTrace();  
 }catch(BadPaddingException e){  
 e.printStackTrace();  
 } catch (InvalidKeyException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
  
}

//DESD.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.DESDController">  
 <children>  
 <Label fx:id="questionDesD" layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="400.0" text="Please Enter CipherText" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="28.0" />  
 </font></Label>  
 <TextField fx:id="DesTextFieldD" layoutX="116.0" layoutY="185.0">  
 <font>  
 <Font size="30.0" />  
 </font>  
 </TextField>  
 <Button fx:id="submitDesD" layoutX="259.0" layoutY="262.0" mnemonicParsing="false" onAction="#submitDesD" text="Submit">  
 <font>  
 <Font size="19.0" />  
 </font>  
 </Button>  
 <Label fx:id="CipherTextDesD" layoutX="110.0" layoutY="99.0" prefHeight="56.0" prefWidth="400.0" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="28.0" />  
 </font>  
 </Label>  
 </children>  
</AnchorPane>

//DESDController.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
  
import javax.crypto.\*;  
import java.security.InvalidKeyException;  
import java.security.NoSuchAlgorithmException;  
import java.util.Arrays;  
  
  
//https://mkyong.com/java/jce-encryption-data-encryption-standard-des-tutorial/  
//Parts of the code was derived from the website above.  
  
public class DESDController {  
  
 //Declaring a variety of buttons, labels, text fields for the user to see  
  
 @FXML  
 private Label questionDesD;  
  
 @FXML  
 private Label CipherTextDesD;  
  
 @FXML  
 private Button submitDesD;  
  
 @FXML  
 private TextField DesTextFieldD;  
  
 String inputDesD;  
  
 public void submitDesD(ActionEvent event){  
 try{  
 //This is a key generator instance imported from the DES library to generate a unique key.  
 KeyGenerator keyGenerator = KeyGenerator.getInstance("DES");  
 SecretKey myDesKey = keyGenerator.generateKey();  
 Cipher desCipher;  
  
  
  
 //Constructs the Cipher from the imported instance  
 desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");  
  
 desCipher.init(Cipher.ENCRYPT\_MODE, myDesKey);  
  
  
 //Gathers the input from the Textfield and storing the input into a String variable  
 String inputDesD = DesTextFieldD.getText();  
  
 //Storing the converted contents from the input into byte array  
 byte[] bytesInput = inputDesD.getBytes();  
  
 byte [] inputEncrypted = desCipher.doFinal(bytesInput);  
  
  
 //Initializing the Cipher for decryption function  
 desCipher.init(Cipher.DECRYPT\_MODE, myDesKey);  
  
  
 // Calling the function to decrypt the input  
 byte[] inputDecrypted = desCipher.doFinal(inputEncrypted);  
  
  
 //Outputting the decrypted data by changing the question label to the output  
 questionDesD.setText("Decryption: " + ((inputEncrypted)));  
 CipherTextDesD.setText("Original Text: " + new String(inputDecrypted));  
  
 // A variety of exceptions including on the library  
 }catch(NoSuchAlgorithmException e){  
 e.printStackTrace();  
 }catch(NoSuchPaddingException e){  
 e.printStackTrace();  
 } catch(IllegalBlockSizeException e){  
 e.printStackTrace();  
 }catch(BadPaddingException e){  
 e.printStackTrace();  
 } catch (InvalidKeyException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
  
}

//Hill.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.HillController">  
 <children>  
 <Label fx:id="questionHill" alignment="CENTER" layoutX="4.0" layoutY="14.0" prefHeight="56.0" prefWidth="592.0" text="Please Enter 2 Character Plaintext" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="28.0" />  
 </font></Label>  
 <Button fx:id="submit" layoutX="269.0" layoutY="261.0" mnemonicParsing="false" onAction="#submit" text="Submit">  
 <font>  
 <Font size="19.0" />  
 </font>  
 </Button>  
 <Label fx:id="key" layoutX="14.0" layoutY="297.0" prefHeight="56.0" prefWidth="174.0" text="Key: HILL" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="28.0" />  
 </font>  
 </Label>  
 <TextField fx:id="HillTextField" layoutX="131.0" layoutY="144.0" prefHeight="56.0" prefWidth="358.0" promptText="Please type only 2 characters" />  
 </children>  
</AnchorPane>

//HillController.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
  
import java.util.List;  
import java.util.Locale;  
  
  
public class HillController {  
 // Declaring labels, textfields, and buttons for the user to interact with  
  
 @FXML  
 private Label questionHill;  
 @FXML  
 private Label keyHill;  
  
 @FXML  
 private Button submit;  
  
 @FXML  
 private TextField blockSize;  
  
 @FXML  
 private TextField HillTextField;  
  
 String input;  
 int block;  
  
  
 public void submit(ActionEvent event) {  
 // An alphabet array for the indexes  
 String[] alphabet = {"A","B","C","D","E","F","G","H", "I", "J", "K", "L", "M", "N", "O", "P","Q", "R", "S", "T","U", "V", "W", "X", "Y", "Z"};  
  
 //Getting the input from the textfield and storing it into a string variable while also forcing the string to be capitalized  
 String input = HillTextField.getText().toUpperCase();  
  
 //Storing the input into a character array so that the indexes could be accessed  
 char[] alpha = input.toCharArray();  
  
 //Getting the length of the input from the textfield  
 int len = input.length();  
  
 int[] arr = new int [len];  
  
 // Setting the values of the inverse key for the calculating with the input indexes  
 int[][] key ={{7,8}  
 , {11,11}};  
  
  
 int[] result = new int[len];  
  
  
 //A for loop subtracting each ASCii value by 65 to get the index and storing those values  
 for(int i = 0; i < len; i++){  
 arr[i] = alpha[i] - 65;  
 }  
  
  
 //Does the calculations of matrix multiplication  
 for(int j = 0; j < len; j++){ // row  
 for(int k = 0; k < len; k++){ // column  
 result[j] += (key[j][k] \* arr[k]);  
  
 }  
  
  
  
 }  
 //Storing the resulted values into an empty string for easier output  
 String strH = "";  
 for(int i = 0; i < len; i++){  
 strH += alphabet[result[i] % 26];  
 }  
 questionHill.setText("Output: " + strH);  
 }  
}

//HillD.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.test2.HillControllerD">  
 <children>  
 <Label fx:id="questionHillD" alignment="CENTER" layoutX="110.0" layoutY="43.0" prefHeight="56.0" prefWidth="381.0" text="Please Enter 2 Character Ciphertext" textAlignment="CENTER">  
 <font>  
 <Font name="Andale Mono" size="17.0" />  
 </font></Label>  
 <TextField fx:id="HillTextFieldD" layoutX="116.0" layoutY="185.0">  
 <font>  
 <Font size="30.0" />  
 </font>  
 </TextField>  
 <Button fx:id="submit" layoutX="259.0" layoutY="262.0" mnemonicParsing="false" onAction="#submit" text="Submit">  
 <font>  
 <Font size="19.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

//HillControllerD.java

package com.example.test2;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
  
import java.util.List;  
import java.util.Locale;  
  
  
public class HillControllerD {  
 // Declaring labels, textfields, and buttons for the user to interact with  
  
 @FXML  
 private Label questionHillD;  
 @FXML  
 private Label keyHillD;  
  
 @FXML  
 private Button submit;  
  
 @FXML  
 private TextField HillTextFieldD;  
  
 String inputD;  
  
  
  
 public void submit(ActionEvent event) {  
 // An alphabet array for the indexes  
 String[] alphabet = {"A","B","C","D","E","F","G","H", "I", "J", "K", "L", "M", "N", "O", "P","Q", "R", "S", "T","U", "V", "W", "X", "Y", "Z"};  
  
 //Getting the input from the textfield and storing it into a string variable while also forcing the string to be capitalized  
 String inputD = HillTextFieldD.getText().toUpperCase();  
  
 //Storing the input into a character array so that the indexes could be accessed  
 char[] alpha = inputD.toCharArray();  
  
 //Getting the length of the input from the textfield  
 int len = inputD.length();  
  
 //Initializing a new array with the same length as the input to store input indexes  
 int[] arr = new int [len];  
  
 // Setting the values of the inverse key for the calculating with the input indexes  
 int[][] inverseKey ={{25,22}  
 , {1,23}};  
  
 //Initializing a new array with the same length as the input to store indexes after calculations  
 int[] result = new int[len];  
  
  
 //A for loop subtracting each ASCii value by 65 to get the index and storing those values  
 for(int i = 0; i < len; i++){  
 arr[i] = alpha[i] - 65;  
 }  
  
  
 //Does the calculations of matrix multiplication  
 for(int j = 0; j < len; j++){ // row  
 for(int k = 0; k < len; k++){ // column  
 result[j] += (inverseKey[j][k] \* arr[k]);  
  
 }  
  
  
  
 }  
  
 //Storing the resulted values into an empty string for easier output  
 String strH = "";  
 for(int i = 0; i < len; i++){  
 strH += alphabet[result[i] % 26]; // The result will reference the alphabet string to gather the appropriate letter  
 System.out.println();  
 }  
 questionHillD.setText("Output: " + strH);  
 }  
}