TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ

KHOA CÔNG NGHỆ THÔNG TIN

Môn: Bảo Mật Thông Tin

Bài thực hành số 1

Bài 1: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa Ceasar.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép ghi File và mở File.

Hướng dẫn mã dịch chuyển Caesar:

-Ta lần lượt đánh chỉ số cho các chữ cái bắt đầu từ 0.

- Gọi k là 1 số nguyên từ 0 ->25 được gọi là khóa.

-Hàm mã hóa: E(p,k)=(p+k)mod26 với p là chỉ số của ký tự cần mã hóa.

-Hàm giải mã: D(c,k)=|c-k|mod26 với c là chỉ số của ký tự cần giải mã.

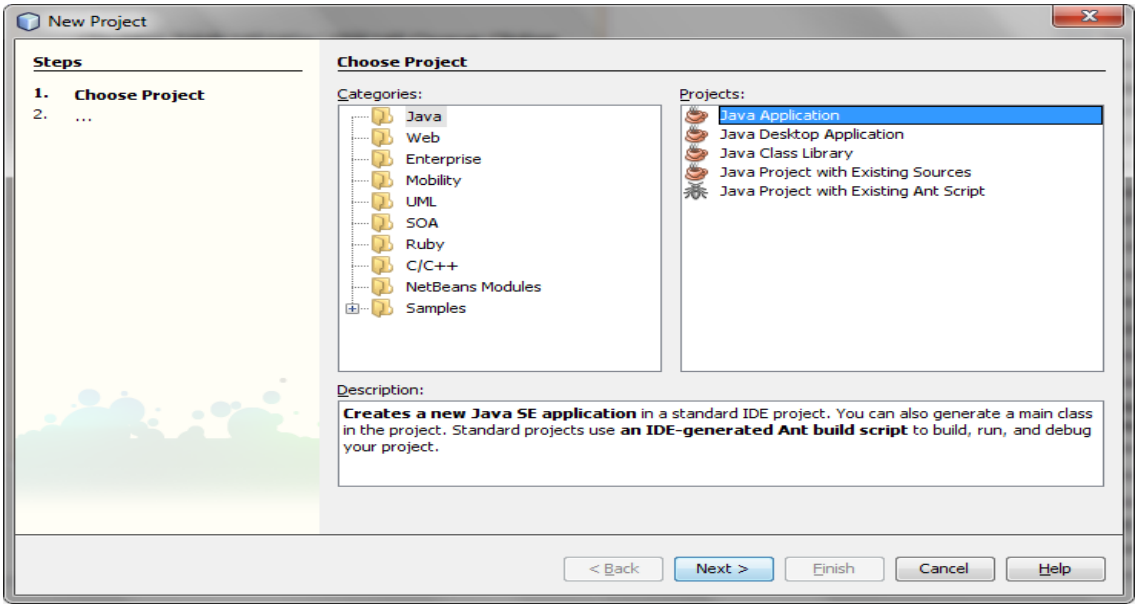
Encryption:

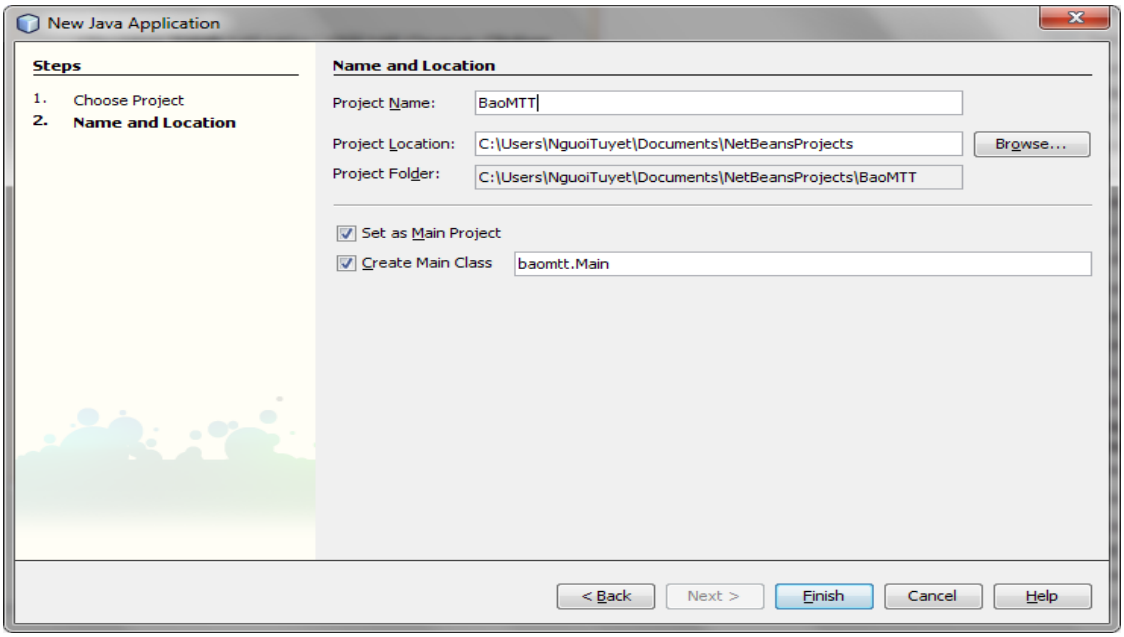
1. Map the plaintext characters to numbers : a = 0, …, z = 25
2. Encrypt the message (sequence of numbers m) using
   $c = a*m + b \mod 26$
   where a and b are the encryption keys.
3. Map the numbers back to characters to obtain the cipher

Decryption:

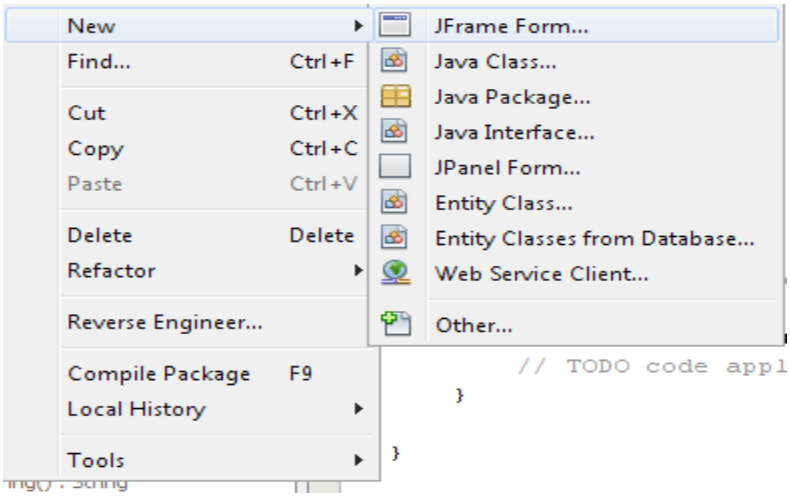
1. Map the cipher characters to numbers : a = 0, …, z = 25
2. Decrypt the number sequence c using
   $m = a^{-1}*c + a^{-1}*b \mod 26$
   where a^{-1} is the multiplicative inverse of a mod 26
3. Map the numbers back to characters

Bước 1: Tạo project mới: File → New Project



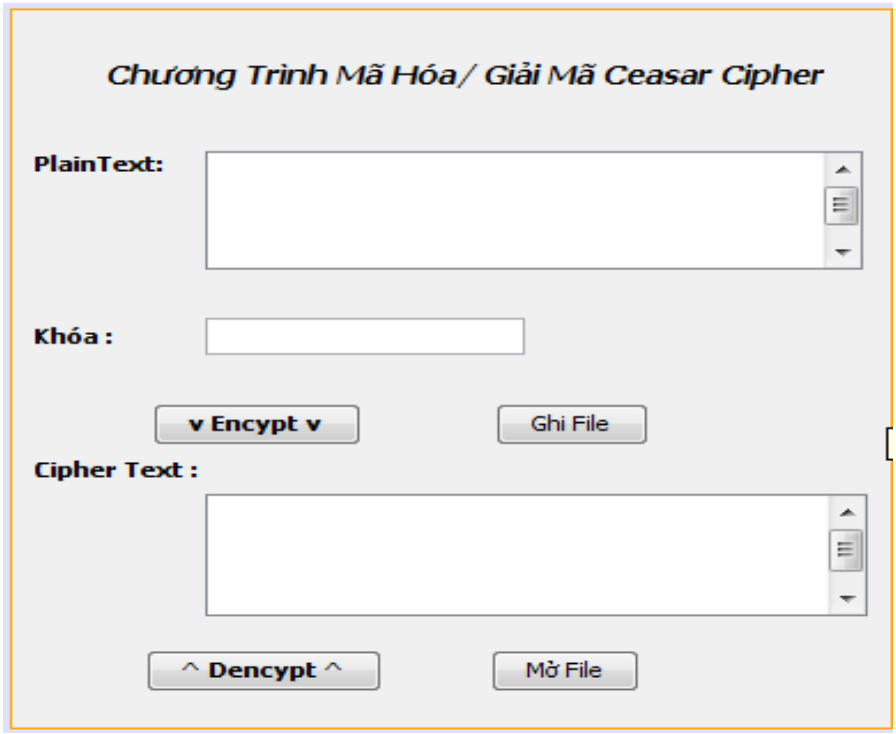


Bước 2: Tạo mới jFrame Form thiết kế:



Giao diện thiết kế Frame:

Bước 3: Thiết kế Form



Bước 4: Viết hàm xữ lý sự kiện

a. Hàm xữ lý sự kiện Encypt

```
private void bntmahoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String plainText=txtvanban.getText();
    String khoa=txtkhoa.getText();
    int key;
    //chuyển kiểu chuỗi thành kiểu int
    key=Integer.parseInt(khoa);
    if(key < 0)
    {
        key = 26 - (-key % 26);
    }
    String result = "";
    for(int i=0 ; i<plainText.length();i++)
    {
        char ch = mahoa(plainText.charAt(i),key);
        result  = result + ch;
    }
    txtmahoa.setText(result);
}
```

b. Hàm xữ lý sự kiện Ghi File

```
private void bntGhiFileActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
         BufferedWriter bw = null;
        //Nơi lưu dữ liệu
        String fileName = "D:\\Dulieu.txt";
        //luu van ban
        String s = txtmahoa.getText();
        // Ghi dữ Liệu S vào tạo tin fileName
        bw = new BufferedWriter(new FileWriter(fileName));
        bw.write(s);
        bw.close();
        JOptionPane.showMessageDialog(null, " Đã Ghi File Thành Công !!!");
    } catch (IOException ex) {
        Logger.getLogger(Ceasar.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

c. Hàm xữ lý sự kiện Dencypt

```
 private void bntgiaimaActionPerformed(java.awt.event.ActionEvent evt) {
     // TODO add your handling code here:
      String cipherText=txtvanban.getText();
     String khoa=txtkhoa.getText();
     int key;
     key=Integer.parseInt(khoa);
     //key từ 0->25 là 1 số nguyên
     if(key < 0)
     {
         key = 26 - (-key % 26);
     }
     String result = "";
     //mã hóa các ký tự của văn bản
     for(int i=0 ; i<cipherText.length();i++)
     {
         char ch = mahoa(cipherText.charAt(i),key);
         result  = result + ch;
     }
     //hiển thị văn bản đã mã hóa
     txtmahoa.setText(result);
 }
//hàm mã hóa : mahoa(ch,k)=(ch+k)mod26 với ch là chỉ số của ký tự cần mã hóa
private char mahoa(char ch, int key)
{
        if( Character.isLetter(ch)){
        ch = (char) ( 'A' +(Character.toUpperCase(ch)-'A'+ key)%26);
    }
    return ch;
}
/**
```

d. Hàm xữ lý sự kiện Mở File

```
private void bntMoFileActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
        // Lớp BufferedReader kế thừa từ lớp Reader thuộc nhóm Input
        //Nhóm này là 1 trong 2 loại chính của Các luồng Ký Tự
        // Có chức năng đọc dữ liệu dạng ký tự
        BufferedReader br = null;
        String fileName = "D:\\Dulieu.txt";
        br = new BufferedReader(new FileReader(fileName));
        //nhận dữ liệu
        StringBuffer sb = new StringBuffer();
        JOptionPane.showMessageDialog(null, " Đã mở File Thành Công !!!");
        //Đọc mỗi lần tối đa 5 ký tự
        char[] ca = new char[5];
        while (br.ready()) {
            int len = br.read(ca);
            sb.append(ca, 0, len);
        }
        br.close();
```

```
        //xuat chuoi
        System.out.println("Du Lieu la :" + " " + sb);
        String chuoi = sb.toString();
        // Hiển thị lên Form
        txtvanban.setText(chuoi);
    } catch (IOException ex) {
        Logger.getLogger(Ceasar.class.getName()).log(Level.SEVERE, null, ex);

    }
```

Bài 2: Viết chương trình mã hóa và giãi mã văn bản với thuật toán mã hóa Vigenere.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng Dẫn: Mật mã Vigenere còn gọi là mật mã nhiều bảng mã. Ưu điểm của mã này là việc sử dụng 26 bảng mã khác nhau. Do đó mà không bị phá trong một thời gian dài. Ngoài ra mã này còn hỗ trợ việc sử dụng từ khóa vô cùng tiện lợi.

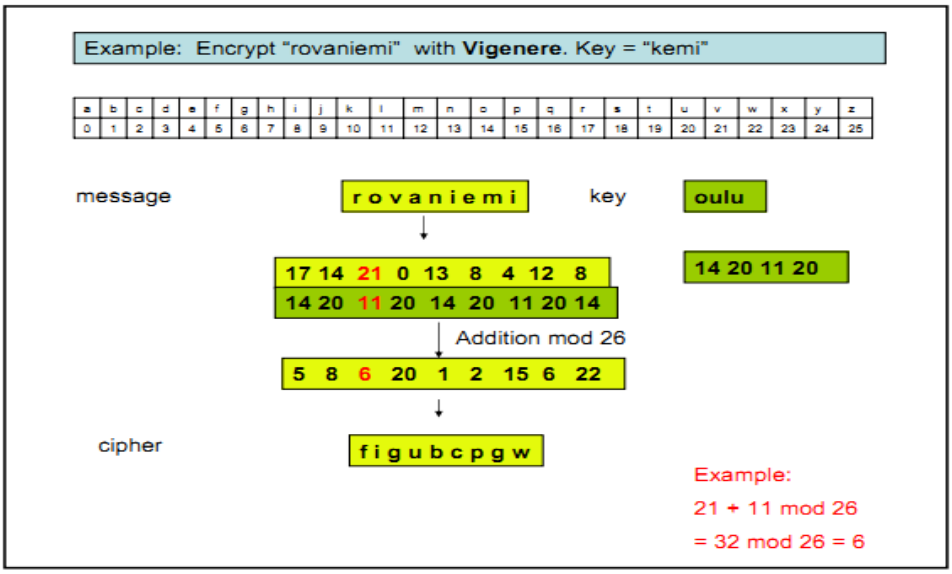
Thuật toán:

- Khoá K là một bộ gồm nhiều khoá con
  - $K = (k_1, k_2, \ldots, k_m)$
- **Mã hoá:**
  - $e_K(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$
- **Giải mã:**
  - $d_K(y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m)$
  - *(cộng, trừ theo modulo 26)*
    - → *"MÃ KHỐI" (block cipher)*

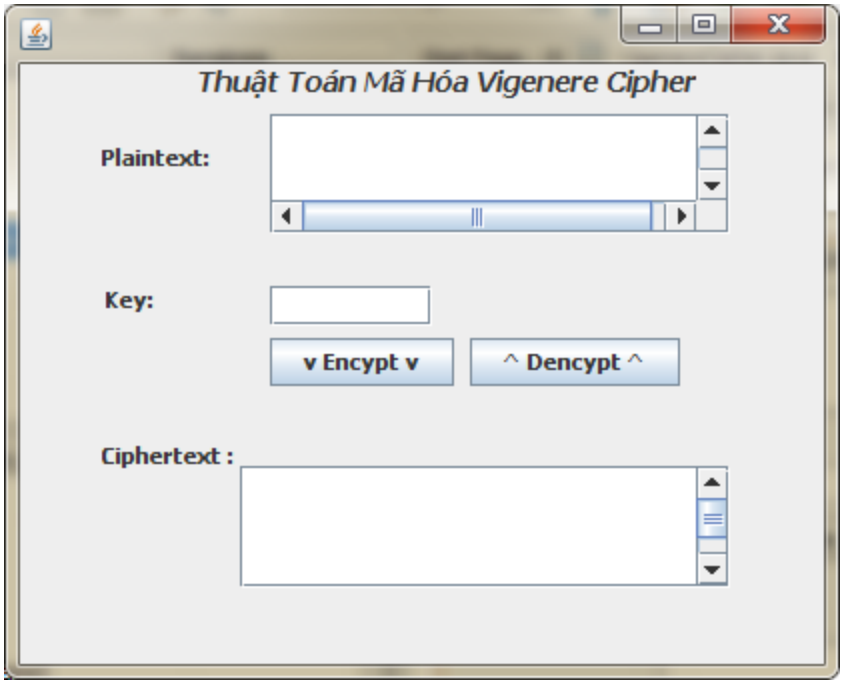
Ví dụ:

- {A, B, C, …, X, Y, Z} = $\mathbf{Z}_{26}$={0, 1, …, 25}.
- K = (2, 8, 15, 7, 4, 17) ("CIPHER").
- p = "thiscryptosy".

- c = “VPXZGIAXIVWP”



Bước 1: Thiết Kế Form :



Bước 2: Viết hàm xử lý sự kiện

a. Hàm xữ lý sự kiện Encypt

```
private void bntmahoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int tableRowSize = 26;
    int tableColumnSize = 26;
    int vignereTable[][] = new int[26][26];

    for (int rows = 0; rows < tableRowSize; rows++){

       for (int columns = 0; columns < tableColumnSize; columns++){

            vignereTable[rows][columns] = (rows + columns) % 26;

         }
     }
     // nhập văn bản cần mã hóa
    System.out.println("Enter the plaintext");
    String plainText= txtvanban.getText();
   // chuyển văn bản sang chữ hoa
    plainText = plainText.toUpperCase();
    // nhập khóa K
    System.out.print("Enter the key: ");
String key = txtkhoa.getText();
key = key.toUpperCase();
String cipherText = "";
int keyIndex = 0;
for (int ptextIndex = 0; ptextIndex < plainText.length(); ptextIndex++)
{
    char pChar = plainText.charAt(ptextIndex);
   int asciiVal = (int) pChar;
    if (pChar == ' '){
         cipherText += pChar;
         continue;
  }
  if (asciiVal < 65 || asciiVal > 90){
       cipherText += pChar;
       continue;
  }
  int basicPlainTextValue = ((int) pChar) - 65;
  char kChar = key.charAt(keyIndex);
  int basicKeyValue = ((int) kChar ) - 65;
  int tableEntry = vignereTable[basicPlainTextValue][basicKeyValue];
```

```
            char cChar = (char) (tableEntry + 65);
            cipherText += cChar;
            keyIndex++;
            if (keyIndex == key.length())
                  keyIndex = 0;
        }
        System.out.println(" cipher text is "+cipherText);
        txtmahoa.setText(cipherText.toString().toUpperCase());

    }
```

b. Hàm xữ lý sự kiện Dencypt

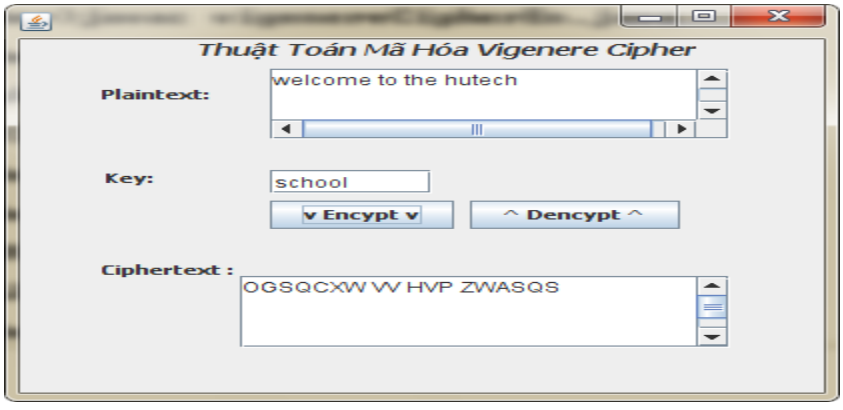
```
    // TODO add your handling code here:
    int tableRowSize = 26;
    int tableColumnSize = 26;
    int vignereTable[][] = new int[26][26];
    String cipherText=txtmahoa.getText();
    String plainText="";
      for (int rows = 0; rows < tableRowSize; rows++){

        for (int columns = 0; columns < tableColumnSize; columns++){
            vignereTable[rows][columns] = (rows + columns) % 26;

         }
    }
    System.out.println("Enter the cipher text");
    cipherText = cipherText.toUpperCase();
    System.out.print("Enter the key: ");
    String key = txtkhoa.getText();
    key = key.toUpperCase();
    int keyIndex = 0;
```

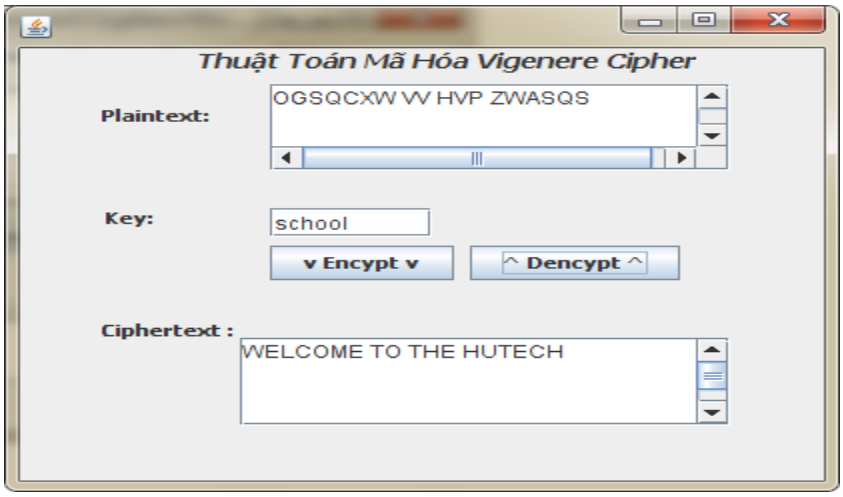
```
for (int ctextIndex = 0; ctextIndex < cipherText.length(); ctextIndex++){
   char cChar = cipherText.charAt(ctextIndex);
   int asciiVal = (int) cChar;
   if (cChar == ' '){
         plainText += cChar;
         continue;
   }
   if (asciiVal < 65 || asciiVal > 90){
       plainText += cChar;
       continue;
   }
    int basiccipherTextValue = ((int) cChar) - 65;
   char kChar = key.charAt(keyIndex);
   int basicKeyValue = ((int) kChar ) - 65;
   for (int pIndex = 0; pIndex < tableColumnSize; pIndex++){
      if (vignereTable[basicKeyValue][pIndex] == basiccipherTextValue){
            char potcChar = (char) (vignereTable[basicKeyValue][pIndex] + 65);
            char potpChar = (char) (pIndex + 65);
            plainText += potpChar;
      }
   }
```

keyIndex ++;

```
    if (keyIndex == key.length())
     keyIndex = 0;

    }

    System.out.println(" plain text is "+plainText);
    txtvanban.setText(cipherText.toString().toUpperCase());
    txtmahoa.setText(plainText.toString().toUpperCase());


}
```

Kết quả:





Bài Tập Về Nhà: Yêu cầu viết phần mềm mã hóa và giải mã với 2 thuật toán trên bao gồm:

- Menu mã hóa: Thuật toán Ceasar, Thuật Toán Vigenere

- Menu giải mã: Thuật toán Ceasar, Thuật Toán Vigenere

- Các chức năng mã hóa và giải mã đều phải có mục mã hóa và giải mã theo File(File có thể là .txt, .dat,…)

Bài 3: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa Rail Fence.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

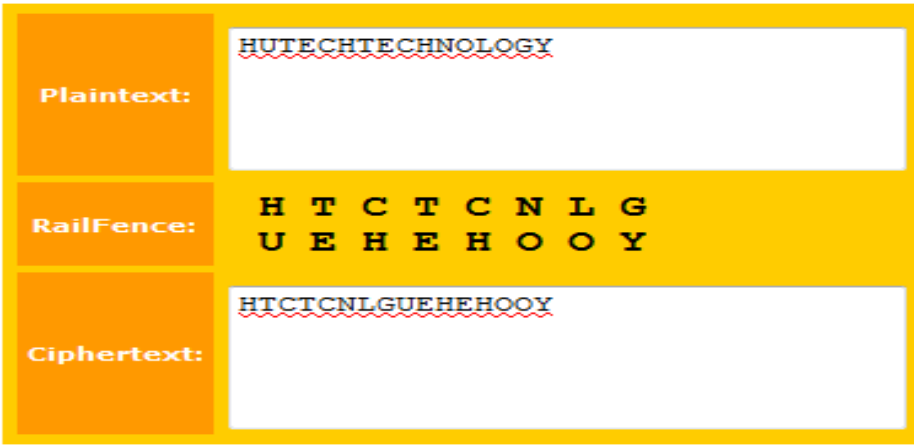
Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

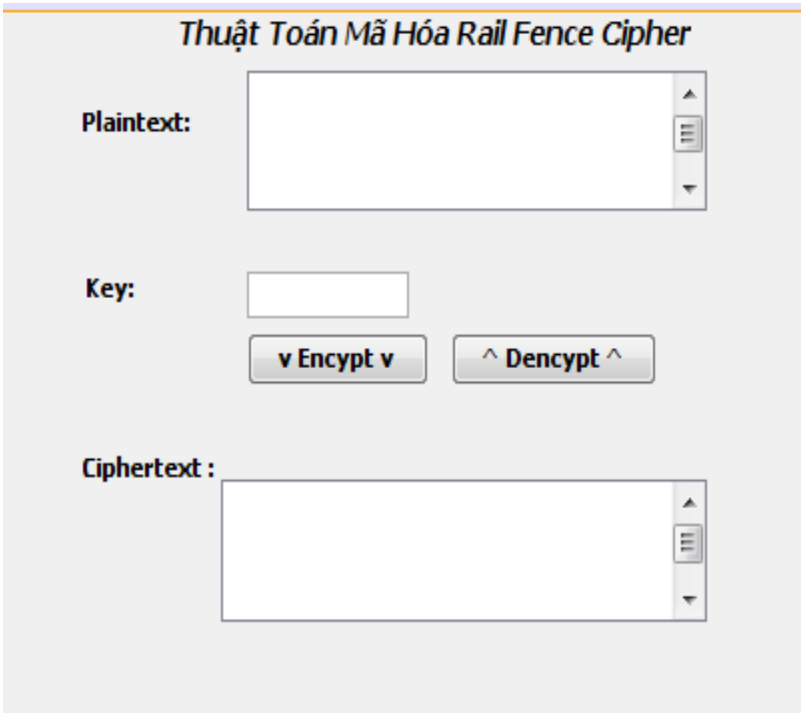
Hướng Dẫn :Mã Rail Fence còn được gọi là mã zig zag là một hình thức của mã chuyển vị:

- Thông điệp được viết lần lượt từ trái qua phải trên các cột (rail) của một hàng dào tưởng tượng theo đường chéo từ trên xuống dưới.
- Theo đường chéo từ dưới lên khi đạt tới cột thấp nhất.

- Và khi đạt tới cột cao nhất, lại viết theo đường chéo từ trên xuống. Cứ lặp đi lặp lại như thế nào cho đến khi viết hết toàn bộ nội dung của thông điệp.
- Ví dụ: mã hóa chuỗi HUTECH TECHNOLOGY với khóa là 2.



Bước 1: Thiết Kế Form :



Bước 2: Viết hàm xử lý sự kiện

a. Hàm xữ lý sự kiện Encypt

```
private void bntmahoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
                    String plainText=txtvanban.getText();
                    int i=0,j,l,r,s,c=0,t=0;
                    String cipherText="";
                    System.out.println("Enter the Rails:");
                    String khoa=txtkhoa.getText();
                    r=Integer.parseInt(khoa);
                   // chuyển kiểu chuỗi sang kiểu int
                    l=plainText.length()/r;
                    s=2*l;
                    char ct[][]=new char [r][s];
                    while(i<r)
                    {
                             ct[i][c]=plainText.charAt(t);
                             t++;
                             if(i==r-1&&t<=plainText.length()-1)
                                 {
                                      c=c+1;
                                      for(j=r-2;j>=0;j--)
                    {
                             ct[j][c]=plainText.charAt(t);
                             t++;
                             if(j==0&&t<=plainText.length()-1)
                             {
                                      c++;
                                      i=0;
                             }
                    }
          }
          i=i+1;
}
for(i=0;i<r;i++)
{
          for(j=0;j<ct[i].length;j++)
                {
            char d=ct[i][j];
            if(d=='\0')
            {
                     continue;
```

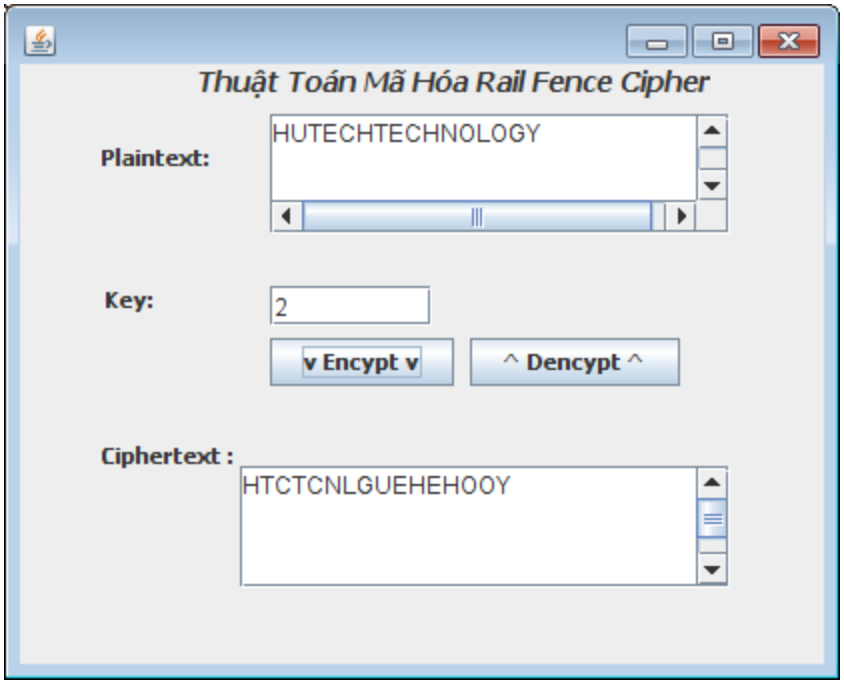
```
                                        }
                                        else
                                        {
                                                cipherText=cipherText+ct[i][j];
                                        }
                                }
                        }
                        System.out.println("The Cipher text is:\n"+cipherText);
                        System.out.println(cipherText.length());
                        txtmahoa.setText(cipherText.toString().toString());

    }
```

b. Hàm xữ lý sự kiện Dencypt

```
private void btDecryptActionPerformed(java.awt.event.ActionEvent evt) {
if(txtClearText.getText().equals("")){
    JOptionPane.showMessageDialog(null,"Clear Text is EMpty");

}else {
    if(txtKey.getText().equals("")){
    JOptionPane.showMessageDialog(null,"Key is EMpty");
    }else{
        String text=txtClearText.getText().trim();
        int key=Integer.parseInt(txtKey.getText());

        String result="";

       int k=0;
       int s=(text.length()/key)+1;

       char ct[][]=new char[key][s];

    for (int i=0;i<key;i++){
        for(int j=0;j<s;j++){
            if(k>=text.length()){
                break;
            }
            ct[i][j]=text.charAt(k);
            k=k+1;
            System.out.print(ct[i][j]+ "  ");

        }
        System.out.println(" ");

    }

     for (int i=0;i<s;i++){
        for(int j=0;j<key;j++){
            if(ct[j][i]==0)
                result=result+'\0';
```

```
                else
               // JOptionPane.showMessageDialog(null,ct[j][i]);
                result=result+ ct[j][i];

            }

        }
       String s1="";
        for(k=0;k<result.length();k++){

            if (((int)result.charAt(k))!=0)
                s1=s1+result.charAt(k);

        }

txtResult.setText(s1);

    }
```

Bước 3: Kiểm Tra



Bài 4: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa PLayFail.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng dẫn:

- **Phương pháp** là lập ma trận 5x5 dựa trên từ khóa cho trước và các ký tự trên bảng chữ cái :

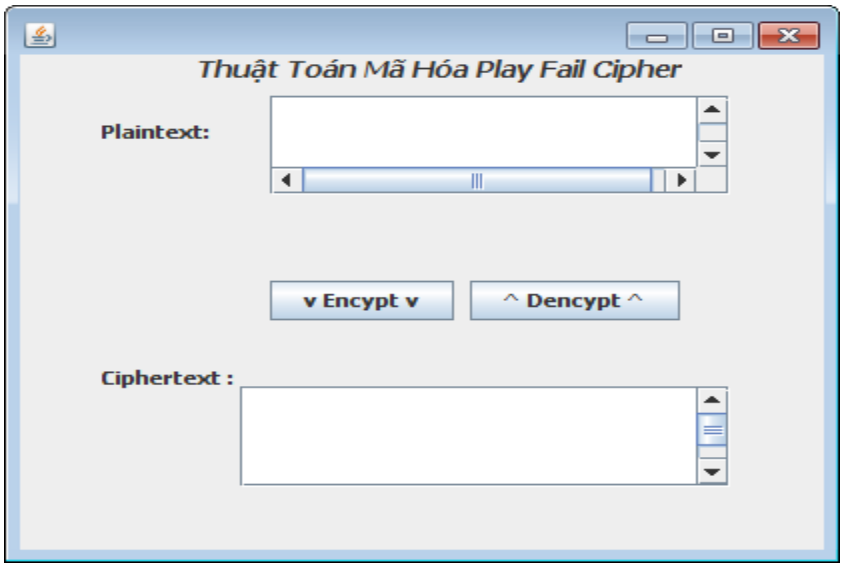
  - Trước hết viết các chữ của từ khoá vào các hàng của ma trận bắt từ hàng thứ nhất.
  - Nếu ma trận còn trống, viết các chữ khác trên bảng chữ cái chưa được sử dụng vào các ô còn lại. Có thể viết theo một trình tự qui ước trước, chẳng hạn từ đầu bảng chữ cái cho đến cuối.
  - Vì có 26 chữ cái tiếng Anh, nên thiếu một ô. Thông thường ta dồn hai chữ nào đó vào một ô chung, chẳng hạn I và J.
  - Giả sử sử dụng từ khoá MORNACHY. Lập ma trận khoá Playfair tương ứng như sau:

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

- **Quy tắc mã hóa và giải mã**
  - Chia bản rõ thành từng cặp chữ. Nếu một cặp nào đó có hai chữ như nhau, thì ta chèn thêm một chữ lọc chẳng hạn X. Ví dụ, trước khi mã **"balloon"** biến đổi thành **"ba lx lo on".**
  - Nếu cả hai chữ trong cặp đều rơi vào cùng một hàng, thì mã mỗi chữ bằng chữ ở phía bên phải nó trong cùng hàng của ma trận khóa (cuộn vòng quanh từ cuối về đầu), chẳng hạn "**ar**" biến đổi thành "RM" .

- Nếu cả hai chữ trong cặp đều rơi vào cùng một cột, thì mã mỗi chữ bằng chữ ở phía bên dưới nó trong cùng cột của ma trận khóa (cuộn vòng quanh từ cuối về đầu), chẳng hạn "**mu**" biến đổi thành "CM".
- Trong các trường hợp khác, mỗi chữ trong cặp được mã bởi chữ cùng hàng với nó và cùng cột với chữ cùng cặp với nó trong ma trận khóa. Chẳng hạn, **"hs"** mã thành **"BP"**, và **"ea"** mã thành **"IM"** hoặc **"JM"** .

Bước 1: Thiết Kế Form :



Bước 2: Viết hàm xử lý sự kiện

a. Hàm xữ lý sự kiện Encypt

```
    private void bntmahoaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
{
  String plainText="",plainTxt, cipherText="";
  String playFairMatrix[][]= {
                                {"M", "O", "N", "A", "R"},
                                {"C", "H", "Y", "B", "D" },
                                {"E","F","G","I","K"},
                                {"L","P","Q","S","T"},
                                {"U","V","W","X","Z"},

                                        };
System.out.println("Enter a plaintext:");
plainTxt = txtvanban.getText();
String temp="";
String arr[]=plainTxt.split(" ");
for(int j=0;j<arr.length;j++)
{
          temp=arr[j];
          if(temp.length()%2!=0)
          {
                    temp=temp+"x";
          }
          plainText=plainText+ temp+" ";

}

for(int i=0; i<plainText.length(); i+=2)
{
    char c = plainText.charAt(i);
          char  d=plainText.charAt(i);
          if(i+1<plainText.length())
          {
                              d = plainText.charAt(i+1);
          }

          String val = String.valueOf(c);
          String vald = String.valueOf(d);
          String index1,index2;
          if(val.equals(" "))
          {
                    cipherText=cipherText+" ";
                    i--;
                    continue;
```

```
            }
            else
            {
                    if(val.equalsIgnoreCase("J"))//to insert the index position for char J
            {
                    index1 = findIndex(playFairMatrix, String.valueOf("I"));
                    }
else
{
        index1 = findIndex(playFairMatrix, String.valueOf(plainText.charAt(i)));
}
if(vald.equalsIgnoreCase("J"))//to insert the index position for char J
{
        index2 = findIndex(playFairMatrix, String.valueOf("I"));

}
else
{
        index2 = findIndex(playFairMatrix, String.valueOf(plainText.charAt(i+1)));
}

if(index1.charAt(0) == index2.charAt(0))//row same
{
       int m = Integer.parseInt(String.valueOf(index1.charAt(1)));//column of index 1
```

```
        int n = Integer.parseInt(String.valueOf(index2.charAt(1)));//column of index 2
        int o = Integer.parseInt(String.valueOf(index1.charAt(0)));//row of index 1
        int p = Integer.parseInt(String.valueOf(index2.charAt(0)));//row of index 2
        if(m==4)
        {
                m=-1;
        }
        if(n==4)
        {
                n=-1;
        }
                cipherText=cipherText+playFairMatrix[o][m+1];
                cipherText=cipherText+playFairMatrix[p][n+1];
}
else if(index1.charAt(1) == index2.charAt(1))//column sname
{ int o = Integer.parseInt(String.valueOf(index1.charAt(0)));//row of index 1
        int m = Integer.parseInt(String.valueOf(index1.charAt(1)));//column of index 1
        int p = Integer.parseInt(String.valueOf(index2.charAt(0)));//row of index 2
        int n = Integer.parseInt(String.valueOf(index2.charAt(1)));//column of index 2
        if(p>3)
        {
                p=-1;
        }
        if(o>3)
        {
                o=-1;
        }
        cipherText=cipherText+playFairMatrix[o+1][m];
        cipherText=cipherText+playFairMatrix[p+1][n];
}
else
{
        int o = Integer.parseInt(String.valueOf(index1.charAt(0)));//row of index 1
         int m = Integer.parseInt(String.valueOf(index1.charAt(1)));//column of index 1
                        int p = Integer.parseInt(String.valueOf(index2.charAt(0)));//row of index
                        int n = Integer.parseInt(String.valueOf(index2.charAt(1)));//column of in
                        cipherText=cipherText+playFairMatrix[o][n];
                        cipherText=cipherText+playFairMatrix[p][m];

                 }
                 }

        }
        System.out.println("The cipher text of the above plain text is:");
        System.out.println(cipherText);
        txtmahoa.setText(cipherText.toString().toUpperCase());

        }
    }
```

b. Hàm xữ lý sự kiện Dencypt

```
    private void bntgiaimaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
  String plainText="",cipherText="";
    String playFairMatrix[][]=              {
                                        {"M", "O", "N", "A","R"},
                                        {"C","H", "Y", "B", "D" },
                                        {"E", "F" ,"G", "I", "K" },
                                        {"L", "P", "Q", "S", "T"},
                                        {"U", "V", "W", "X","Z"},
                                        };
System.out.println("Enter a Ciphertext:");
cipherText = txtmahoa.getText();
txtmahoa.setText(cipherText.toString().toUpperCase());
for(int i=0; i<cipherText.length(); i+=2)
{
            char c = cipherText.charAt(i);
            char  d=cipherText.charAt(i);
            if(i+1<cipherText.length())
            {
                    d = cipherText.charAt(i+1);
            }
            String val = String.valueOf(c);
            String vald = String.valueOf(d);
            String index1,index2;
            if(val.equals(" "))
            {
        plainText=plainText+" ";
        i--;
        continue;
}
else
     if(val.equalsIgnoreCase("J"))//to insert the index position for char J
        {
                  index1 = findIndex(playFairMatrix, String.valueOf("I"));

        }
        else
        {
         index1 = findIndex(playFairMatrix, String.valueOf(cipherText.charAt(i)));
        }
        if(vald.equalsIgnoreCase("J"))//to insert the index position for char J
        {
                   index2 = findIndex(playFairMatrix, String.valueOf("I"));

        }
        else
        {
         index2 = findIndex(playFairMatrix, String.valueOf(cipherText.charAt(i+1)));
          }
   if(index1.charAt(0) == index2.charAt(0))//row same
   {
           int m = Integer.parseInt(String.valueOf(index1.charAt(1)));//column of index 1
           int n = Integer.parseInt(String.valueOf(index2.charAt(1)));//column of index 2
```

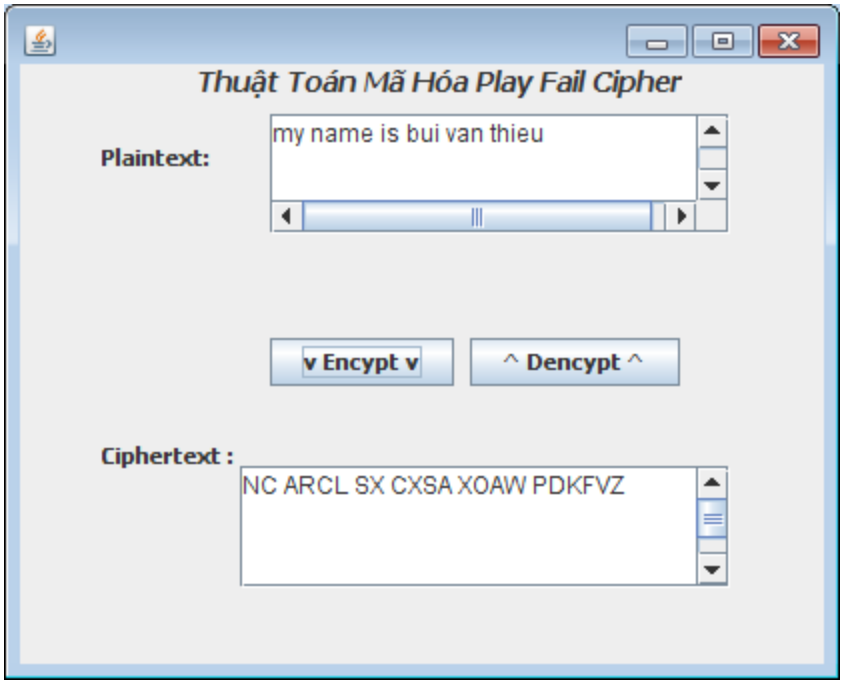
```
            int o = Integer.parseInt(String.valueOf(index1.charAt(0)));//row of index 1
            int p = Integer.parseInt(String.valueOf(index2.charAt(0)));//row of index 2
            if(m==0)
            {
                    m=5;
            }
            if(n==0)
            {
                    n=5;
            }
                    plainText=plainText+playFairMatrix[o][m-1];
                    plainText=plainText+playFairMatrix[p][n-1];
}
 else if(index1.charAt(1) == index2.charAt(1))//column sname
 {
            int o = Integer.parseInt(String.valueOf(index1.charAt(0)));//row of index 1
            int m = Integer.parseInt(String.valueOf(index1.charAt(1)));//column of index 1
            int p = Integer.parseInt(String.valueOf(index2.charAt(0)));//row of index 2
            int n = Integer.parseInt(String.valueOf(index2.charAt(1)));//column of index 2
            if(p==0)
            {
                    p=5;
            }
            if(o==0)
            {
                    o=5;
            }
                    plainText=plainText+playFairMatrix[o-1][m];
                    plainText=plainText+playFairMatrix[p-1][n];
        }
       else
          {
                    int o = Integer.parseInt(String.valueOf(index1.charAt(0)));//row of index 1
                    int m = Integer.parseInt(String.valueOf(index1.charAt(1)));//column of index
                    int p = Integer.parseInt(String.valueOf(index2.charAt(0)));//row of index 2
                    int n = Integer.parseInt(String.valueOf(index2.charAt(1)));//column of index
                    plainText=plainText+playFairMatrix[o][n];
                    plainText=plainText+playFairMatrix[p][m];
        }
        }


System.out.println("plaintext text of the above cipher text is:");
System.out.println(plainText);
txtmahoa.setText(plainText.toString().toUpperCase());
```

}

c. Hàm FindIndex

```
public static String findIndex(String[][] arr, String test)
        {
                String index = "";
                for(int i=0; i<arr.length; i++)
                {
                        for(int j=0; j<arr[i].length; j++)
                        {
                                if(test.equalsIgnoreCase(arr[i][j]))
                                {
                                        index = String.valueOf(i) + String.valueOf(j);
                                        return index;
                                }
                        }
                }
                return null;
        }
```

Bước 3: Kết quả



Bài 5: Viết chương trình mã hóa và giãi mã văn bản với thuật toán mã hóa Transposition cipher.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng dẫn :<u>Hệ mã hóa đổi chỗ (Transposition Cipher)</u>

Là hệ mã hóa trong đó các kí tự của bản gốc được giữ nguyên, nhưng vị trí bị thay đổi.

Đảo ngược toàn bộ plaintex : nghĩa là bản gốc được viết theo thứ tự ngược lại từ sau ra trước.

Ví dụ Plaintext: SECURE EMAIL

Bản mã: LIAMEERUCES

Mã hóa theo mẫu hình học: bản gốc được sắp xếp lại theo một mẫu hình học nào đó, thường là một mảng hoặc ma trận hai chiều.

Ví dụ: bản gốc ban đầu là BAO MAT

Ví dụ mã hoá theo mẫu hình học.

Cột 1	Cột 2	Cột 3
B	A	O
M	A	T

Nếu lấy các cột theo thứ tự 2, 3, 1. Bản mã sẽ là AAOTBM

Đổi chỗ cột: đổi chỗ các kí tự trong plaintext thành dạng hình chữ nhật theo cột.

Ví dụ : Bản gốc BAO MAT THU DIEN TU

Bản gốc được chuyển thành ma trận 3x5 như sau:

Bảng ví dụ mã hóa bằng phương pháp đổi chỗ cột

Cột 1	Cột 2	Cột 3	Cột 4	Cột 5
B	M	T	D	N
A	A	H	I	T
O	T	U	E	U

Vì có 5 cột nên chúng có thể được sắp lại theo 5! = 120 cách khác nhau

Nếu ta chuyển vị các cột theo thứ tự 3, 5, 2, 4, 1 rồi lấy các kí tự theo hàng ngang ta sẽ thu được bản mã: TNMDBHTAIAUUTEO.

***Hoán vị các kí tự của bản gốc theo chu kỳ cố định d*:** Nếu hàm f là hoán vị của một khối gồm d kí tự thì khóa mã hóa được biểu diễn bởi K(d, f)

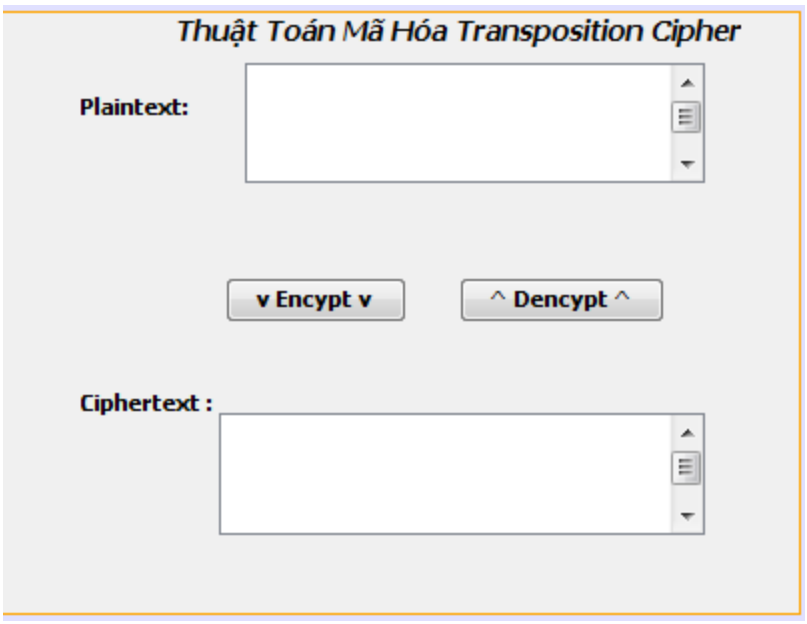
Ví dụ: với d = 5, f hoán vị của dãy 12345 thành 35142

Bảng Hoán vị các kí tự của bản gốc theo chu kỳ cố định d

Vị trí ban đầu	Vị trí hoán vị	Nội dung mã hóa	Mã hóa
1	3	G	O
2	5	R	P
3	1	O	G
4	4	U	U
5	2	P	R

Theo bảng trên bản gốc ban đầu được mã hóa thành OPGUP.

Bước 1: Thiết Kế Form :



Bước 2: Viết hàm xử lý sự kiện

a. Hàm xử lý sự kiện Encypt

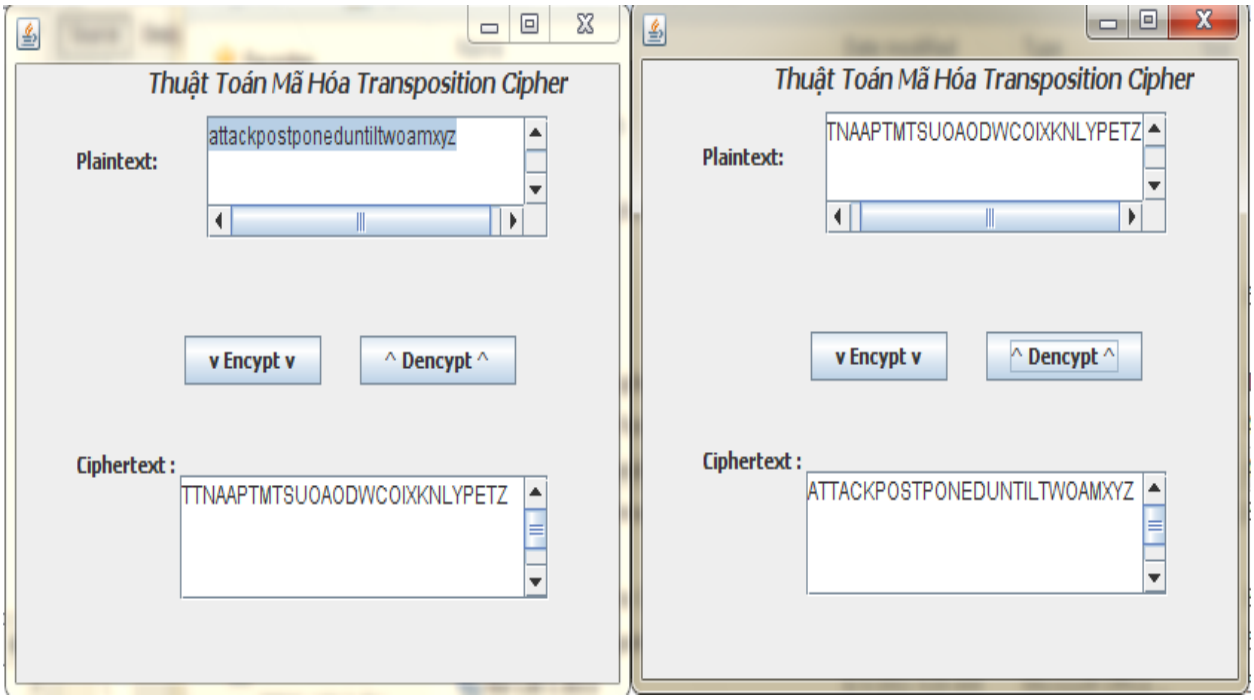
```
private void bntmahoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
             char letter[]=new char [25];
                          String plainText,cipherText="",jpt="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
             int c=0,l=1,v=0,i=0,t=1,vc=0,r=0;
             char  mat[][]=new char[50][50];
             int key[]={4,3,1,2,5,6,7};
            char  z;
            System.out.println("Enter the plain text:");
            plainText=txtvanban.getText();
            for(i=0;i<c+1;i++)
            {
             for(int j=0;j<key.length;j++)
             {
               if(c<plainText.length())
             {
                       char a=plainText.charAt(c);
                       mat[i][j]=a;
                       c++;
                       v=i+1;  //for row count
             }
             if(c==plainText.length()&&j!=vc)//to last string X Y Z..
             {
                       r=key.length-j;
                       if(r<=jpt.length())
                       {
                               z=jpt.charAt(jpt.length()-r);
                               mat[i][j+1]= z;
```

```
                    }
                }
            }
            //them doan nay
            t=2;
            if(c==plainText.length()&&r==0)// thoat vong lap
                {i=c+2;
                }
            }
               c=1;
                .
             while(c<=key.length)

             {
                 //het
               for(i=0;i<key.length;i++)
              {
                      if(key[i]==c)//to match the sequence of key position
                      {
                                int k=0;
                                for(l=0;l<v;l++)
                                {
                                         cipherText=cipherText+mat[k][i];
                                        k++;//to increase the row
                                }
                      }
              }
               c++;
                    }
                    cipherText=cipherText.toUpperCase();
                    System.out.println("\nThe Cipher Text is:\n"+cipherText);
                    txtmahoa.setText(cipherText.toString().toUpperCase());
    }
```

b. Hàm xử lý sự kiện Dencypt

```
private void bntgiaimaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
                    String plainText="",cipherText="",jpt="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
            int c=0,l=1,v=0,i=0,j=0,k=0;
            char  mat[][]=new char[50][50];
            char  pmat[][]=new char[50][50];
            int key[]={4,3,1,2,5,6,7};
            char  z;
            System.out.println("Enter the Cipher text:");
            cipherText=txtmahoa.getText();
            txtvanban.setText(cipherText.toString().toUpperCase());
            v=(cipherText.length()/key.length);
            while(c<key.length)
            {
                     for(j=0;j<key.length;j++)
                     {
                             if(key[j]==c+1)
                             {
                                     for(i=0;i<v;i++)
                                     {
                                             mat[i][j]=cipherText.charAt(k);
                                             k++;
                                     }
                             }
                    }
                     c++;
            }
             for(int p=0;p<4;p++)
             {
                      for(int q=0;q<key.length;q++)
                      {
                              plainText=plainText+mat[p][q];
                     }
             }
             System.out.println("\nThe Plain text is:\n"+plainText);
             txtmahoa.setText(plainText.toString().toUpperCase());


}
```

Bước 3: Kết Quả



Bài Tập: Viết phần mềm mã hóa văn bãn với các thuật toán mã hóa trên.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Cho phép bên gũi mã hóa dữ liệu và bên nhận mã hóa dữ liệu với khóa K.