Bài 4: Viết chương trình mã hóa và giãi 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

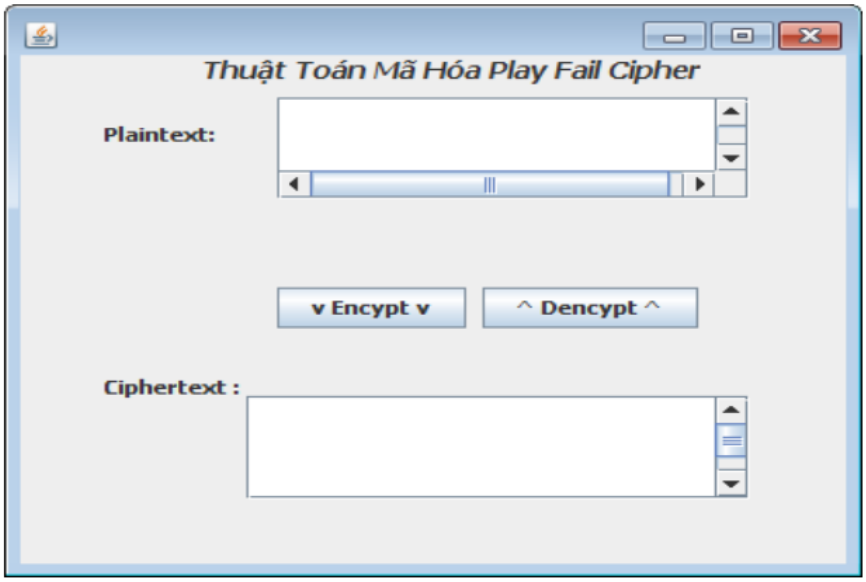
Quy tắc mã hóa và giải mã

- 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