

```

import java.util.*;
class Playfair{
static Scanner sc=new Scanner(System.in);
static String
pattern="1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ",key,plain="",cipher="";
static char[][] mat=new char[6][6];
static ArrayList<String> slice=new ArrayList<String>(); static ArrayList<String>
enc_slice=new ArrayList<String>();
public static void main(String[] a) throws Exception
{
System.out.println("Enter choice, Encryption 'E' or Decryption 'D'");
char choice = (char)System.in.read();

        if(keycheck()==true)
            switch(choice)
{
case 'E': {System.out.println("Ready for Encryption"); encrypt();}
            break;
case 'D': {System.out.println("Ready for Decryption"); decrypt();}
            break;
default : System.out.println("Choose a valid option");
}

        else keycheck();
}

static void encrypt() throws Exception
{
    System.out.println("Enter plain text to encrypt, valid symbols are: A-Z and 0-9");
    plain=sc.next();
    split(plain);encode();
    System.out.println("Cipher text is: "+cipher);
}

static void decrypt() throws Exception
{
    System.out.println("Enter cipher text to decrypt, valid symbols are: A-Z and 0-9");
    cipher=sc.next();
    split(cipher);decode();
    System.out.println("Plain text is: "+plain);
}

static void matrix(String k) throws Exception
{
    char[] arr=pattern.toCharArray();
    int l=0,t=0;
    for(int i=0; i<6;i++)
    {for(int j=0; j<6;j++)
    {

        if(l<k.length())
        {
            mat[i][j]=k.charAt(l);
            l++; continue;
        }

        while(true)
        {
            if(!k.contains(arr[t]+""))

```

```

        {mat[i][j] = arr[t]; t++; break;}
    else {t++; continue;}
    }
    }
}

System.out.println("Press Y to print matrix");
if((char)System.in.read()=='Y')
{
    System.out.println("Generated matrix is: ");

    for (int i = 0; i < mat.length; i++) {
        for (int j = 0; j < mat[i].length; j++) {
            System.out.print(mat[i][j] + " ");
        }
        System.out.println();
    }
}

static boolean keycheck() throws Exception
{int i,j=0,f=0;
System.out.println("Enter the key, valid symbols are: A-Z and 0-9 without
repeataations");
key = sc.next();

for(i=0;i<key.length();i++)
    {for(j=i+1;j<key.length();j++)
        if(key.charAt(i)==key.charAt(j)) f=1;
    }

    for(i=0;i<key.length();i++)
    {
        if(pattern.contains(key.charAt(i)+""))
            continue;
        else break;}

    if((i==key.length())&&(f==0))
    {matrix(key);return true;}
    else
    {System.out.println("Entered key: "+key+" is not accepted, try
again.");
        return false;}
}

static void split(String text) throws Exception
{
    int i;
    for(i=1;i<text.length();i)
    {if(text.charAt(i-1)!=text.charAt(i))
        {slice.add(""+text.charAt(i-1)+text.charAt(i));
        i+=2;}
        else {slice.add(""+text.charAt(i-1)+'X');
        i+=1;}
    }

    if((i%2==1)&&(i==text.length())){slice.add(""+text.charAt(i-1)+'X');}

    System.out.println("Press Y to print Input slices");
    if((char)System.in.read()=='Y')
    {
        System.out.println("Input text slices are:");
    }
}

```

```

for (i = 0; i < slice.size(); i++) {
    System.out.print(slice.get(i) + " ");
}
System.out.println();
}
}

static void encode() throws Exception
{int[] pos0=new int[slice.size()];int[] pos1=new int[slice.size()];
    for (int i = 0; i < mat.length; i++) {
        for (int j = 0; j < mat[i].length; j++) {
            for(int k=0; k<slice.size();k++)
            {
                if(slice.get(k).charAt(0)==mat[i][j])
                    pos0[k]=6*i+j;
                if(slice.get(k).charAt(1)==mat[i][j])
                    pos1[k]=6*i+j;
            }
        }
        //System.out.println();
    }
    for(int k=0; k<slice.size();k++)
    {int row0=pos0[k]/6, col0=pos0[k]%6;
    int row1=pos1[k]/6, col1=pos1[k]%6;

        if(row0==row1)
            enc_slice.add(""+mat[row0][(col0+1)%6]+mat[row1][(col1+1)%6]);

        else if(col0==col1)
            enc_slice.add(""+mat[(row0+1)%6][col0]+mat[(row1+1)%6][col1]);
        else enc_slice.add(""+mat[row0][col1]+mat[row1][col0]);
    }

    System.out.print("Encoded slices are: ");
    for (int i = 0; i < enc_slice.size(); i++) {
        System.out.print(enc_slice.get(i) + " ");
        cipher=cipher+enc_slice.get(i);
    }
}

static void decode() throws Exception
{int[] pos0=new int[slice.size()];int[] pos1=new int[slice.size()];
    for (int i = 0; i < mat.length; i++) {
        for (int j = 0; j < mat[i].length; j++) {
            for(int k=0; k<slice.size();k++)
            {
                if(slice.get(k).charAt(0)==mat[i][j])
                    pos0[k]=6*i+j;
                if(slice.get(k).charAt(1)==mat[i][j])
                    pos1[k]=6*i+j;
            }
        }
        //System.out.println();
    }
    for(int k=0; k<slice.size();k++)
    {int row0=pos0[k]/6, col0=pos0[k]%6;
    int row1=pos1[k]/6, col1=pos1[k]%6;

```

```

        if(row0==row1)
            enc_slice.add(""+mat[row0][(col0-1+6)%6]+mat[row1][(col1-1+6)%6]);

        else if(col0==col1)
            enc_slice.add(""+mat[(row0-1+6)%6][col0]+mat[(row1-1+6)%6][col1]);
        else enc_slice.add(""+mat[row0][col1]+mat[row1][col0]);
    }
    System.out.print("Decoded slices are: ");
    for (int i = 0; i < enc_slice.size(); i++) {
        System.out.print(enc_slice.get(i) + " ");
        plain=plain+enc_slice.get(i);
    }
}
}

```