**EXPERIMENT - I**

**SUBSTITUTION CIPHERS**

**1a) Caesar cipher**

**Aim**: -To encrypt and decrypt text using Caesar cipher

**Procedure**: -

Step 1: - Create a class with all necessary variables

Step 2: - Ask whether to encrypt or decrypt

Step 3: - Get the string from user

Step 4: - If encryption ask key of choice

Step 5: - Use the encryption technique a=a+x where x,is the key of choice

Step 6: - Return back to start of alphabetical order if a reaches 27

Step 7: - If decryption, brute force for all valid keys and extract from one of them.

**Source code**: -

import java.util.\*;

public class encdec

{

public static void main(String arg[])

{

Scanner a = new Scanner(System.in);

int x,ax,z,i,b,j;

char ch,cha,bx,c,al;

String enc;

String dec;

String text1 = "";

do

{

System.out.println("\tENTER CHOICE");

System.out.println("\t 1.ENCRYPT");

System.out.println("\t 2.DECRYPT");

ch=a.next().charAt(0);

switch(ch)

{

case '1' :

System.out.println("\tENTER STRING TO ENCRYPT");

enc=a.next();

System.out.println("\tENTER KEY OF CHOICE");

x=a.nextInt();

for( i=0; i < enc.length();i++)

{

al = enc.charAt(i);

al = (char) (al + x);

if(al > 'z')

{

al = (char) (al+'a'-'z'-1);

}

text1 = text1 + al;

}

System.out.println("YOUR TEXT="+text1);

break;

case '2':

System.out.println("\tENTER STRING TO DECRYPT");

dec=a.next();

for(j=1;j<26; j++)

{

String text="";

for( i=0; i < dec.length();i++)

{

char alp = dec.charAt(i);

alp = (char) (alp - j);

if(alp < 'a')

{

alp = (char) (alp+('z'-'a'+1));

}

text = text + alp;

}

System.out.println("your text for your key"+j+"is:-\t"+text);

}

break;

}

System.out.println("wanna continue?? -> y/n");

cha=a.next().charAt(0);

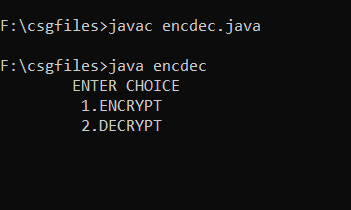
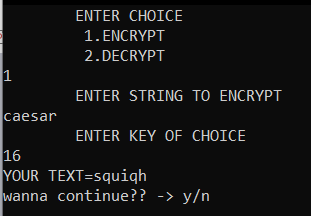
}

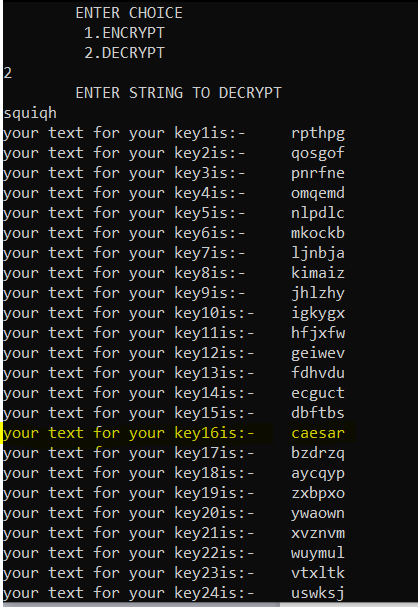
while(cha=='y');

}

}

**Output**: -



**1b) Affine cipher**

**Aim**: -To encrypt and decrypt text using Affine cipher

**Procedure**: -

Step 1: - Create a class with all necessary variables

Step 2: - Get the string from user

Step 3: - If encryption ask key of choice

Step 4: - Use the encryption technique E ( x ) = ( a x + b ) mod m

modulus m: size of the alphabet,a and b: key of the cipher and x is the letter

Step 5: - Decrypt using given key

**Source code**: -

import java.io.\*;

import java.util.\*;

class afine

{

static int a = 17;

static int b = 20;

int g,h;

static String encryptMessage(char[] msg)

{

Scanner m =new Scanner(System.in);

int g,h;

System.out.println("ENTER KEY");

g=m.nextInt();

h=m.nextInt();

String cipher = "";

for (int i = 0; i < msg.length; i++)

{

if (msg[i] != ' ')

{

cipher = cipher

+ (char) ((((a \* (msg[i] - 'A')) + b) % 26) + 'A');

}

else

{

cipher += msg[i];

}

}

return cipher;

}

static String decryptCipher(String cipher)

{

String msg = "";

int a\_inv = 0;

int flag = 0;

for (int i = 0; i < 26; i++)

{

flag = (a \* i) % 26;

if (flag == 1)

{

a\_inv = i;

}

}

for (int i = 0; i < cipher.length(); i++)

{

if (cipher.charAt(i) != ' ')

{

msg = msg + (char) (((a\_inv \*

((cipher.charAt(i) + 'A' - b)) % 26)) + 'A');

}

else

{

msg += cipher.charAt(i);

}

}

return msg;

}

public static void main(String[] args)

{

String msg = "";

Scanner o = new Scanner(System.in);

msg=o.next();

String cipherText = encryptMessage(msg.toCharArray());

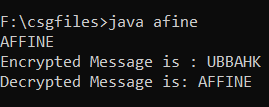
System.out.println("Encrypted Message is : " + cipherText);

System.out.println("Decrypted Message is: " + decryptCipher(cipherText));

}

}

**Output: -**



**1c)Playfair cipher**

**Aim: - To encrypt text using playfair cipher**

**Procedure**: -

Step 1: - Create a class with all necessary variables

Step 2: - Get the string from user

Step 3: - If encryption ask key of choice

Step 4: - Use the encryption technique E[I]=P[I]+K[I] where E is the encrypted text,P being the plain-text and K being key.

Step 5: - Decrypt using given key

**Source code: -**

**(encryption)**

import java.util.Scanner;

public class encPlay{

static void encrypt(String pt, String key) {

char[] pt\_arr=pt.toCharArray();

char[] key\_arr=key.toCharArray();

char[] alpha={'A','B','C','D','E','F','G','H','I','K','L',

'M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};

char[][] table=new char[5][5];

char[] encpt\_arr = new char[100];

int encpt\_arr\_ctr=0;

int ctr=-1;

for(int i=0;i<key\_arr.length;i++)

for(int j=0;j<25;j++)

if(key\_arr[i]==alpha[j])

{

ctr++;

int round = ctr/5;

table[round][ctr%5]=alpha[j];

alpha[j]='0';

break;

}

for(int i=0;i<alpha.length;i++)

{

if(alpha[i]!='0')

{

ctr++;

int round = ctr/5;

table[round][ctr%5]=alpha[i];

}

}

char[] any=new char[100]; // stores intermediate string

int c=0;

for(int i=0;i<pt\_arr.length;i++)

{

if(i==0 && pt\_arr[i]!=pt\_arr[i+1] &&

i+1<pt\_arr.length)

{

any[c++]=pt\_arr[i];

continue;

}

if(i==1 && pt\_arr[i]!=pt\_arr[i+1] &&

i+1<pt\_arr.length)

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-2 && i%2==0 &&

pt\_arr[i]!=pt\_arr[i-1]

&& pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-2 && i%2==0 &&

pt\_arr[i]!=pt\_arr[i-1]

&& pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i==pt\_arr.length-2 && i%2==0 &&

pt\_arr[i]==pt\_arr[i-1]

&& pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-2 && i%2==0 &&

pt\_arr[i]==pt\_arr[i-1]

&& pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i==pt\_arr.length-2 && i%2!=0 &&

pt\_arr[i]!=pt\_arr[i-1]

&& pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-2 && i%2!=0 &&

pt\_arr[i]!=pt\_arr[i-1]

&& pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i==pt\_arr.length-2 && i%2!=0 &&

pt\_arr[i]==pt\_arr[i-1]

&& pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-2 && i%2!=0 &&

pt\_arr[i]==pt\_arr[i-1]

&& pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i==pt\_arr.length-1 && i%2==0 &&

pt\_arr[i]!=pt\_arr[i-1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-1 && i%2==0 &&

pt\_arr[i]==pt\_arr[i-1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i==pt\_arr.length-1 && i%2!=0 &&

pt\_arr[i]!=pt\_arr[i-1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i==pt\_arr.length-1 && i%2!=0 &&

pt\_arr[i]==pt\_arr[i-1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i%2==0 && pt\_arr[i]!=pt\_arr[i-1] &&

pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i%2==0 && pt\_arr[i]!=pt\_arr[i-1] &&

pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i%2==0 && pt\_arr[i]==pt\_arr[i-1] &&

pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i%2==0 && pt\_arr[i]==pt\_arr[i-1] &&

pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i%2!=0 && pt\_arr[i]!=pt\_arr[i-1] &&

pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

if(i%2!=0 && pt\_arr[i]!=pt\_arr[i-1] &&

pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i%2!=0 && pt\_arr[i]==pt\_arr[i-1] &&

pt\_arr[i]!=pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

continue;

}

if(i%2!=0 && pt\_arr[i]==pt\_arr[i-1] &&

pt\_arr[i]==pt\_arr[i+1])

{

any[c++]=pt\_arr[i];

any[c++]='X';

continue;

}

}

if(c%2!=0)

any[c++]='X';

System.out.print("\nThe Intermediate Text

is: ");

for(int i=0;i<c;i++)

System.out.print(any[i]);

System.out.println();

for(int i=0;i<c;i=i+2)

{

if(any[i]=='J')

any[i]='I';

if(any[i+1]=='J')

any[i+1]='I';

int row1=0,row2=0,col1=0,col2=0;

for(int j=0;j<5;j++)

{

for(int k=0;k<5;k++)

{

if(any[i]==table[j][k])

{

row1=j;

col1=k;

break;

}

}

}

for(int j=0;j<5;j++)

{

for(int k=0;k<5;k++)

{

if(any[i+1]==table[j][k])

{

row2=j;

col2=k;

break;

}

}

}

if(row1==row2)

{

col1=(col1+1)%5;

col2=(col2+1)%5;

encpt\_arr[encpt\_arr\_ctr++]=table[row1][col1];

encpt\_arr[encpt\_arr\_ctr++]=table[row2][col2];

}

else if(col1==col2)

{

row1=(row1+1)%5;

row2=(row2+1)%5;

encpt\_arr[encpt\_arr\_ctr++]=table[row1][col1];

encpt\_arr[encpt\_arr\_ctr++]=table[row2][col2];

}

else if(row1!=row2 && col1!=col2)

{

int row=0,col=0;

row=row1;

col=col2;

encpt\_arr[encpt\_arr\_ctr++]=table[row][col];

row=row2;

col=col1;

encpt\_arr[encpt\_arr\_ctr++]=table[row][col];

}

else

{

}

}

System.out.println("\nThe Reference Table

is: ");

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

System.out.print(table[i][j]+" ");

System.out.print("\n");

}

System.out.print("\nThe Encrypted text is:

");

for(int i=0;i<encpt\_arr\_ctr;i++)

System.out.print(encpt\_arr[i]);

}

public static void main(String[] args) {

String pt, key;

System.out.print("Enter plaintext: ");

Scanner scanner = new Scanner(System.in);

pt=scanner.nextLine();

System.out.print("Enter key: ");

key=scanner.nextLine();

scanner.close();

encrypt(pt,key);

}

}

DECRYPTION

import java.util.Scanner;

public class Main{

static void decrypt(String encpt, String key) {

char[] encpt\_arr=encpt.toCharArray();

char[] key\_arr=key.toCharArray();

char[]

alpha={'A','B','C','D','E','F','G','H','I','K','L',

'M','N','O','P','Q','R','S','T','U','V','W','X'

,'Y','Z'};

char[][] table=new char[5][5];

int ctr=-1;

for(int i=0;i<key\_arr.length;i++)

for(int j=0;j<25;j++)

if(key\_arr[i]==alpha[j])

{

ctr++;

int round = ctr/5;

table[round][ctr%5]=alpha[j];

alpha[j]='0';

break;

}

for(int i=0;i<alpha.length;i++)

{

if(alpha[i]!='0')

{

ctr++;

int round = ctr/5;

table[round][ctr%5]=alpha[i];

}

}

System.out.println("\nThe Reference Tableis:");

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

System.out.print(table[i][j]+" ");

System.out.println();

}

char[] any=new char[encpt\_arr.length];

int any\_ctr=0;

for(int i=0;i<encpt\_arr.length;i=i+2)

{

int row1=0,row2=0,col1=0,col2=0;

for(int j=0;j<5;j++)

{

for(int k=0;k<5;k++)

{

if(encpt\_arr[i]==table[j][k])

{

row1=j;

col1=k;

break;

}

}

}

for(int j=0;j<5;j++)

{

for(int k=0;k<5;k++)

{

if(encpt\_arr[i+1]==table[j][k])

{

row2=j;

col2=k;

break;

}

}

}

if(row1==row2)

{

col1=(col1-1+5)%5;

col2=(col2-1+5)%5;

any[any\_ctr++]=table[row1][col1];

any[any\_ctr++]=table[row2][col2];

}

else if(col1==col2)

{

row1=(row1-1+5)%5;

row2=(row2-1+5)%5;

any[any\_ctr++]=table[row1][col1];

any[any\_ctr++]=table[row2][col2];

}

else if(row1!=row2 && col1!=col2)

{

int row=0,col=0;

row=row1;

col=col2;

any[any\_ctr++]=table[row][col];

row=row2;

col=col1;

any[any\_ctr++]=table[row][col];

}

else

{

}

}

System.out.print("\nThe Intermediate Textis: ");

for(int i=0;i<any\_ctr;i++)

System.out.print(any[i]);

char[] decpt\_arr=new char[100];

int decpt\_arr\_ctr=0;

for(int i=0;i<any\_ctr;i++)

{

if(i==0)

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

if(i==1 && any[i-1]==any[i+1] &&

any[i]=='X')

{

continue;

}

if(i==1 && any[i-1]!=any[i+1] &&

any[i]!='X')

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

if(i==2 && any[i-1]==any[i+1] &&

any[i]=='X')

{

continue;

}

if(i==2 && any[i-1]!=any[i+1] &&

any[i]!='X')

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

if(i!=any\_ctr-2 && i!=any\_ctr-1

&& any[i-1]==any[i+1] && any[i]=='X')

{

continue;

}

if(i!=any\_ctr-2 && i!=any\_ctr-1

&& any[i-1]==any[i+1] && any[i]!='X')

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

if(i==any\_ctr-2 && any[i-1]==any[i+1]

&& any[i]=='X')

{

continue;

}

if(i==any\_ctr-2 && any[i-1]==any[i+1]

&& any[i]!='X')

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

if(i==any\_ctr-1 && i%2!=0 &&

any[i]=='X')

{

continue;

}

if(i==any\_ctr-1 && any[i]!='X')

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

if(i==any\_ctr-1 && i%2==0 &&

any[i]=='X')

{

continue;

}

if(i>=0)

{

decpt\_arr[decpt\_arr\_ctr++]=any[i];

continue;

}

}

System.out.print("\n\nThe Decrypted Textis: ");

for(int i=0;i<decpt\_arr\_ctr;i++)

System.out.print(decpt\_arr[i]);

}

public static void main(String[] args) {

String encpt;

System.out.print("Enter Ciphertext: ");

Scanner scanner = new Scanner(System.in);

encpt = scanner.nextLine();

String key;

System.out.print("Enter Key: ");

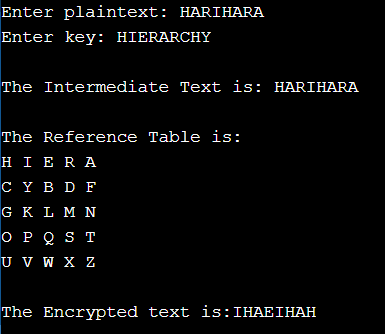
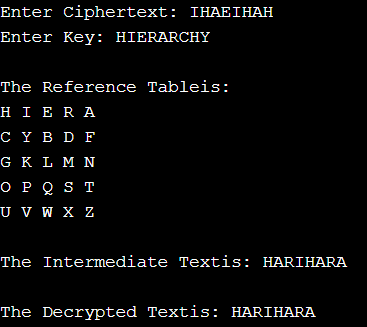
key = scanner.nextLine();

decrypt(encpt,key);

}

}

**Output**

**1d) Multiplicative cipher**

**Aim**: -To encrypt and decrypt text using multiplicative cipher

**Procedure**: -

Step 1: - Create a class with all necessary variables

Step 2: - Get the string from user

Step 3: - If encryption ask key of choice

Step 4: - Use the encryption technique E ( x ) = ( E(x)\*key)%26 )

Step 5: - Decrypt using given key

**Source code**: -

import java.util.\*;

class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int shift,i,n;

String str;

String str1="";

String str2="";

System.out.println("Enter the plaintext");

str=sc.nextLine();

str=str.toLowerCase();

n=str.length();

char ch1[]=str.toCharArray();

char ch3;

char ch4;

System.out.println("Enter the key");

shift=sc.nextInt();

System.out.println();

System.out.println("Encrypted text is");

for(i=0;i<n;i++)

{

if(Character.isLetter(ch1[i]))

{

ch3=(char)(((int)ch1[i]\*shift-97)%26+97);

str1=str1+ch3;

}

else if(ch1[i]==' ')

{

str1=str1+ch1[i];

}

}

System.out.println(str1);

int q=0,flag=0;

for(i=0;i<26;i++)

{

if(((i\*26)+1)%shift==0)

{

q=((i\*26)+1)/shift;

break;

}

}

System.out.println();

System.out.println("Decrypted text is");

char ch2[]=str1.toCharArray();

for(i=0;i<str1.length();i++)

{

if(Character.isLetter(ch2[i]))

{

ch4=(char)(((int)ch2[i]\*q-97)%26+97);

str2=str2+ch4;

}

else if(ch2[i]==' ')

{

str2=str2+ch2[i];

}

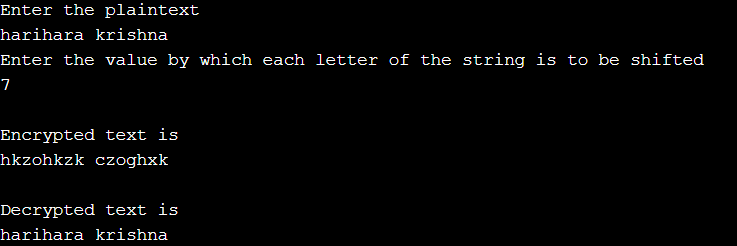
}

System.out.println(str2);

}

}

**Output: -**



**1d) Autokey cipher**

**Aim**: -To encrypt and decrypt text using Autokey cipher

**Procedure**: -

Step 1: - Create a class with all necessary variables

Step 2: - Get the string from user

Step 3: - If encryption ask key of choice

Step 4: - Use the encryption technique E[I]=P[I]+K[I] where E is the encrypted text,P being the plain-text and K being key.

Step 5: - Decrypt using given key

**Source code**: -

import static java.lang.System.in;

import static java.lang.System.out;

import java.util.\*;

public class AutoKeyCipher {

public static Scanner in = new Scanner(System.in);

private static String encryption(String plainText, String keyPhrase) {

String cipherText = "";

plainText = plainText.toUpperCase();

keyPhrase = keyPhrase.toUpperCase();

for (int i = 0; i < plainText.length(); i++) {

int x = (((plainText.charAt(i) - 'A') + (keyPhrase.charAt(i) - 'A')) % 26);

cipherText += (char) (x + 'A');

}

return cipherText;

}

private static String decryption(String cipherText, String keyPhrase) {

String plainText = "";

cipherText = cipherText.toUpperCase();

keyPhrase = keyPhrase.toUpperCase();

for (int i = 0; i < cipherText.length(); i++) {

int x = (((cipherText.charAt(i) - 'A') - (keyPhrase.charAt(i) - 'A')) % 26);

x = (x < 0) ? (26 - Math.abs(x)) : x;

plainText += (char) (x + 'A');

keyPhrase += (char) (x + 'A');

}

return plainText;

}

public static void main(String[] args) {

out.println("if you want encryption press 1,if you want decryption press 2");

int command = in.nextInt();

in.nextLine();

switch (command) {

case 1:

out.println("Please enter your message");

String msgToEencrypt = in.nextLine().replaceAll(" ", "");

out.println("Please enter your key");

String keyToEencrypt = in.nextLine().replaceAll(" ", "");

keyToEencrypt += msgToEencrypt;

out.println(encryption(msgToEencrypt, keyToEencrypt));

break;

case 2:

out.println("Please enter your message");

String msgToDecrypt = in.nextLine();

out.println("Please enter your key");

String keyToDecrypt = in.nextLine();

out.println(decryption(msgToDecrypt, keyToDecrypt));

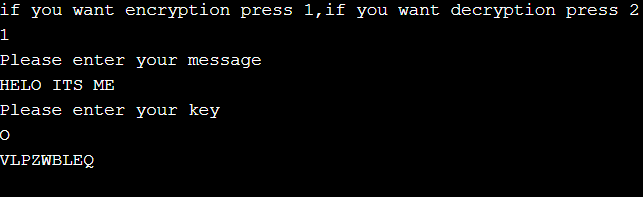
break;

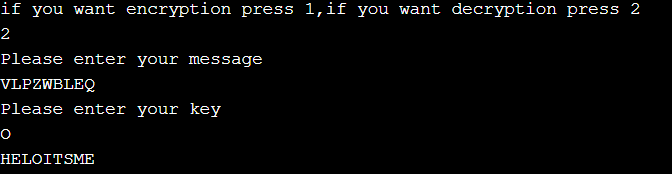
}

}

}

**Output: -**





**1e) Vigenere Cipher**

**Aim: -To encrypt and decrypt using vigenere cipher**

**Source code: -**

import java.util.\*;

public class Main

{

public static char[][] table=new char[26][26];

public static String key=new String("");

static void makeit()

{

int i,j;

for(i=0;i<26;i++)

{

for(j=0;j<i+26;j++)

{ int a=j+65;

if(a>90)

{

table[i][j-i]=(char)(a%90+64);

}

else

{

table[i][j-i]=(char)a;

}

}

}

}

static void make(int v)

{

char ke[]={'A','N','Y'};

for(int i=0;i<v;i++)

{

key=key+Character.toString(ke[i%13]);

}

}

static String ncr(String p)

{

char[] ec=p.toCharArray();

char[] cak=key.toCharArray();

String ec2 = new String("");

//stores row n column values

for(int i=0;i<p.length();i++)

{

int j=cak[i];

int a=j-65;

int b=ec[i];

b=b-65;

ec2=ec2+Character.toString(table[a][b]);

}

return ec2;

}

static String dcr(String p)

{

char[][] d1=key.toCharArray();

char[][] d2=p.toCharArray();

String dp=new String("");

for(int k=0;k<key.length();k++)

{

int i=d1[k],j,b=0;

i=i-65;

for(j=0;j<26;j++)

{

if(table[i][j]==d2[k])

{

b=j;

break;

}

}

dp=dp+Character.toString((char)(b+65));

}

return dp;

}

public static void main(String a[])

{

makeit();

System.out.println("Enter plain text");

Scanner n=new Scanner(System.in);

String p=n.nextLine();

make(p.length());

String e=ncr(p);

System.out.println("Encrypted text"+e);

String d=dcr(p);

System.out.println("decrypted text"+d);

}

}