/\*\*

\* Make OK Cipher work here.

\* @author (your name) Billers and Fredders

\* @version (a version number or a date)

\*/

public class OK{

int a;

int b;

int c;

int d;

int e;

int[]i={1,2,3,4,5};

OK n=new OK(i);

public OK(int[] seed ){

a = seed[0];

b = seed[1];

c = seed[2];

d = seed[3];

e = seed[4];

}

/\* public void printCrap(){

System.out.println(a+"\n"+b+"\n"+c+"\n"+d+"\n"+e);

}\*/

public String garbageAdder(String input){

String returnStuffs = "";

int counter = d;

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

returnStuffs += input.substring(i,i+1);

for(int w = 0; w<counter; w++){

returnStuffs += (char)((int)(Math.random()\*120)+15);

}

counter += e;

}

return returnStuffs;

}

public String garbageSubber(String input1){

String returnStuffs1 = "";

int test = d;

int counter1 = d;

returnStuffs1 += input1.substring(0,1);

for(int i2=1;i2<input1.length();i2++){

if(test == 0){

returnStuffs1 += input1.substring(i2, i2+1);

counter1 += e;

test = counter1;

}

else{

test --;

}

}

return returnStuffs1;

}

public String shift(String input, int num){

String finalValue="";

for(int i3=0;i3<input.length();i3++){

finalValue+=(char)(input.codePointAt(i3)+num);

}

return finalValue;

}

public String AlmostEncrypt(String input2){

String oldText = "";

String newText = "";

String temp = "";

int sum = a+c;

int instaTest = 0;

int instaTests = 0;

oldText += (char)(input2.codePointAt(0)+a);

for(int L=1;L<=b;L++){

//System.out.println(oldText.length());

for(int s = 0; s<oldText.length();s++){

instaTest=oldText.codePointAt(s);

instaTests=(int)(Math.random()\*(instaTest-1))+1;

//temp+=(char)(instaTest-instaTests);

//temp+=(char)(instaTests);

temp += (char)(instaTests+sum)+""+(char)(instaTest-instaTests+sum+c);

sum += c+c;

newText += temp;

temp = "";

}

oldText=newText;

newText="";

}

return oldText;

}

public char convertToLetter(char input){

int finalValue = (int)(input)-65;

if(finalValue<0){

finalValue \*=-1;

finalValue -= 1;

finalValue %= 26;

finalValue = 25-finalValue;}

else{

finalValue = finalValue% 26;

}

finalValue += 65;

return (char)finalValue;

}

public char almostDecrypt(String input3){

int sum2 = 0;

int baseTotal = 0;

int multiplicity = 0;

for(int i4 = b; i4>0; i4--){

sum2 += (int)Math.pow(2.0,(double)i4);

}

for(int i5 = 1; i5<=sum2; i5++){

multiplicity += i5;

}

sum2 = (sum2+1)\*a+multiplicity\*c;

for(int i6 = 0; i6<input3.length(); i6++){

baseTotal+=input3.codePointAt(i6);

}

sum2 = baseTotal - sum2;

return (char)sum2;

}

public String Decrypt(String input4){

String endValue = "";

//String test2 = garbageSubber(input4);

String test2 = input4;

String temp2 = "";

String useAge = "";

final int testValue = (int)(Math.pow(2.0,(double)b));

//System.out.println(test2 + " " + test2.length());

for(int h = 0; h<test2.length(); h+= testValue){

temp2 += ""+input4.substring(h,h+testValue);

useAge += almostDecrypt(temp2);

endValue += "" + useAge;

//System.out.println(endValue);

temp2 = "";

useAge = "";

}

return endValue;

//return test;

}

public String Encrypt(String input5){

String all = "";

for(int t = 0;t<input5.length(); t++){

//System.out.println("CHOC: " + all);

all += "" + AlmostEncrypt(input5.substring(t,t+1));

}

//all = garbageAdder(all);

return all;

}

public void retOK(){

String m=n.garbageAdder(n.Encrypt("WASD"));

System.out.println("Encypted: "+m);

String t=n.Decrypt(n.garbageSubber(m));

System.out.println("Decrypted: "+t);}

}