package ispracs;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.Scanner;

import java.util.Set;

public class Cipher {

static String s;

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the String without spaces:");

s=sc.next();

s=s.toUpperCase();

HashMap<Integer,Character>map=new HashMap<>();

initisalise(map);

Set<Integer>set=map.keySet();

System.out.println("Enter the numeric key for additive substitution cipher:");

int key=sc.nextInt();

int arr[]=new int[s.length()];

conversion(arr,map,set);

String cipher=substitution(key,arr,map,set);

System.out.println("------------------------");

System.out.println("Substituition cipher:-");

System.out.println(cipher);

System.out.println("------------------------");

System.out.println("This cipher goes as input to transposition cipher");

System.out.println("Enter the alphabetic key for transposition :");

String key2=sc.next();

int key2arr[]=new int[key2.length()];

keycal(key2,key2arr);

System.out.println("Key for transposition:-");

for(int i=0;i<key2.length();i++)

System.out.print(key2arr[i]+" ");

System.out.println("");

int rows=(int) Math.ceil((float)cipher.length()/(float)key2.length());

int size= rows\*key2.length();

char transarr[][]=new char[rows+1][key2.length()];

matrix(size,transarr,cipher,key2.length(),rows,key2arr);

String transcipher=displaytrans(transarr,rows,key2.length(),size);

System.out.println("\nTransposition cipher:-"+transcipher);

}

static void initisalise(HashMap<Integer,Character>map) {

char c='A';

for(int i=0;i<=25;i++){

map.put(i,c);

c=(char)(c+1);

}

}

static void conversion(int[] arr, HashMap<Integer, Character> map, Set<Integer> set) {

int pos=-1;

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

char c=s.charAt(i);

for(int k:set){

if(map.get(k)==c){

pos=k;

break;

}

}

arr[i]=pos;

}

}

static String substitution(int key, int[] arr, HashMap<Integer, Character> map, Set<Integer> set) {

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

arr[i]=(arr[i]+key) % 26;

String cipher="";

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

char c=map.get(arr[i]);

cipher=cipher+String.valueOf(c);

}

return cipher;

}

static void keycal(String key2, int[] keyarr) {

key2=key2.toUpperCase();

char[] key2arr=new char[key2.length()];

key2arr=key2.toCharArray();

Arrays.sort(key2arr);

ArrayList<Character>al=new ArrayList<>();

for(char c:key2arr){

al.add(c);

} for(int i=0;i<key2.length();i++){

char c=key2.charAt(i);

int index=al.indexOf(c);

al.remove(index);

al.add(index,'#');

keyarr[i]=index+1;

}

}

static void matrix(int size, char[][] transarr, String cipher,int n,int rows,int[] key2arr) {

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

cipher=cipher+"$";

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

transarr[0][i]=(char)(key2arr[i]+'0');

int k=0;

for(int i=1;i<=rows;i++){

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

{ transarr[i][j]=cipher.charAt(k);

k++;}

}

}

static String displaytrans(char[][] transarr,int rows,int col,int size ) {

System.out.println("Matrix formed:-");

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

System.out.println("");

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

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

}

int k=1;

int colindex=0;

String transcipher="";

while(true){

for(int i=0;i<col;i++){

if((char)(k+'0')==transarr[0][i]){

colindex=i;

k++;

break;

}

}

for(int j=1;j<=rows;j++){

transcipher+=String.valueOf(transarr[j][colindex]);

}

if(size==transcipher.length())

break;

}

return transcipher;

}

}