package myti;

import java.math.BigInteger;

import java.util.Scanner;

public class AffineCipher {

private static int firstKey = 5;

private static int secondKey = 3;

private static int module = 26;

Scanner sc = new Scanner(System.in);

public static void main(String[] args) {

//String input = "abcdefghijklmnopqrstuvwxyz";

System.out.println("please input the message");

Scanner sc = new Scanner(System.in);

String input = sc.next();

String cipher = encrypt(input);

String deciphered = decrypt(cipher);

//System.out.println("Source: " + input);

System.out.println("Encrypted: " + cipher);

System.out.println("Decrypted: " + deciphered);

}

static String encrypt(String input) {

StringBuilder builder = new StringBuilder();

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

char character = input.charAt(in);

if (Character.isLetter(character)) {

character = (char) ((firstKey \* (character - 'a') + secondKey) % module + 'a');

}

builder.append(character);

}

return builder.toString();

}

static String decrypt(String input) {

StringBuilder builder = new StringBuilder();

// compute firstKey^-1 aka "modular inverse"

BigInteger inverse = BigInteger.valueOf(firstKey).modInverse(BigInteger.valueOf(module));

// perform actual decryption

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

char character = input.charAt(in);

if (Character.isLetter(character)) {

int decoded = inverse.intValue() \* (character - 'a' - secondKey + module);

character = (char) (decoded % module + 'a');

}

builder.append(character);

}

return builder.toString();

}

}