描述
  
正整数A和正整数B 的最小公倍数是指 能被A和B整除的最小的正整数值，设计一个算法，求输入A和B的最小公倍数。
  
   
数据范围：1 \le a,b \le 100000 \1≤a,b≤100000
  
输入描述：
  
输入两个正整数A和B。
  
   
输出描述：
  
输出A和B的最小公倍数。
  
   
示例1
  
输入：
  
5 7
  
   
输出：
  
35
  
   
示例2
  
输入：
  
2 4
  
   
输出：
  
4

描述  
计算一个浮点数的立方根，不使用库函数。  
保留一位小数。  
  
数据范围：|val| \le 20 \∣val∣≤20   
  
输入描述：  
待求解参数，为double类型（一个实数）  
  
输出描述：  
输出参数的立方根。保留一位小数。  
  
示例1  
输入：  
19.9  
  
输出：  
2.7  
  
示例2  
输入：  
2.7  
  
输出：  
1.4

描述  
将一个字符串str的内容颠倒过来，并输出。  
  
数据范围：1 \le len(str) \le 10000\1≤len(str)≤10000   
输入描述：  
输入一个字符串，可以有空格  
  
输出描述：  
输出逆序的字符串  
  
示例1  
输入：  
I am a student  
  
输出：  
tneduts a ma I  
  
示例2  
输入：  
nowcoder  
  
输出：  
redocwon

描述  
输入 n 个整型数，统计其中的负数个数并求所有非负数的平均值，结果保留一位小数，如果没有非负数，则平均值为0  
本题有多组输入数据，输入到文件末尾。  
  
数据范围：1 \le n \le 50000 \1≤n≤50000 ，其中每个数都满足 |val| \le 10^{6} \∣val∣≤10   
6  
   
输入描述：  
输入任意个整数，每行输入一个。  
  
输出描述：  
输出负数个数以及所有非负数的平均值  
  
示例1  
输入：  
-13  
-4  
-7  
  
输出：  
3  
0.0  
  
示例2  
输入：  
-12  
1  
2  
  
输出：  
1  
1.5

描述  
Redraiment是走梅花桩的高手。Redraiment可以选择任意一个起点，从前到后，但只能从低处往高处的桩子走。他希望走的步数最多，你能替Redraiment研究他最多走的步数吗？  
  
数据范围：每组数据长度满足 1 \le n \le 200 \1≤n≤200 ， 数据大小满足 1 \le val \le 350 \1≤val≤350   
  
  
输入描述：  
数据共2行，第1行先输入数组的个数，第2行再输入梅花桩的高度  
  
输出描述：  
输出一个结果  
  
示例1  
输入：  
6  
2 5 1 5 4 5   
  
输出：  
3  
  
说明：  
6个点的高度各为 2 5 1 5 4 5  
如从第1格开始走,最多为3步, 2 4 5 ，下标分别是 1 5 6  
从第2格开始走,最多只有1步,5  
而从第3格开始走最多有3步,1 4 5， 下标分别是 3 5 6  
从第5格开始走最多有2步,4 5， 下标分别是 5 6  
所以这个结果是3。

描述  
输入一个只包含小写英文字母和数字的字符串，按照不同字符统计个数由多到少输出统计结果，如果统计的个数相同，则按照ASCII码由小到大排序输出。  
数据范围：字符串长度满足 1 \le len(str) \le 1000 \1≤len(str)≤1000   
  
输入描述：  
一个只包含小写英文字母和数字的字符串。  
  
输出描述：  
一个字符串，为不同字母出现次数的降序表示。若出现次数相同，则按ASCII码的升序输出。  
  
示例1  
输入：  
aaddccdc  
  
输出：  
cda  
  
说明：  
样例里，c和d出现3次，a出现2次，但c的ASCII码比d小，所以先输出c，再输出d，最后输出a.

描述  
输入整型数组和排序标识，对其元素按照升序或降序进行排序  
  
数据范围： 1 \le n \le 1000 \1≤n≤1000 ，元素大小满足 0 \le val \le 100000 \0≤val≤100000   
输入描述：  
第一行输入数组元素个数  
第二行输入待排序的数组，每个数用空格隔开  
第三行输入一个整数0或1。0代表升序排序，1代表降序排序  
  
输出描述：  
输出排好序的数字  
  
示例1  
输入：  
8  
1 2 4 9 3 55 64 25  
0  
  
输出：  
1 2 3 4 9 25 55 64  
  
示例2  
输入：  
5  
1 2 3 4 5  
1  
  
输出：  
5 4 3 2 1

描述  
等差数列 2，5，8，11，14。。。。  
（从 2 开始的 3 为公差的等差数列）  
输出求等差数列前n项和  
  
  
数据范围： 1 \le n \le 1000 \1≤n≤1000   
输入描述：  
输入一个正整数n。  
  
输出描述：  
输出一个相加后的整数。  
  
示例1  
输入：  
2  
  
输出：  
7  
  
说明：  
2+5=7   
示例2  
输入：  
275  
  
输出：  
113575  
  
说明：  
2+5+...+821+824=113575

描述  
自守数是指一个数的平方的尾数等于该数自身的自然数。例如：25^2 = 625，76^2 = 5776，9376^2 = 87909376。请求出n(包括n)以内的自守数的个数  
  
  
数据范围： 1 \le n \le 10000 \1≤n≤10000   
  
  
输入描述：  
int型整数  
  
输出描述：  
n以内自守数的数量。  
  
示例1  
输入：  
6  
  
输出：  
4  
  
说明：  
有0，1，5，6这四个自守数   
示例2  
输入：  
1  
  
输出：  
2  
  
说明：  
有0, 1这两个自守数

描述  
1 总体说明  
考生需要模拟实现一个简单的自动售货系统，实现投币、购买商品、退币、查询库存商品及存钱盒信息的功能。  
系统初始化时自动售货机中商品为6种商品,商品的单价参见1.1规格说明，存钱盒内放置1元、2元、5元、10元钱币，商品数量和钱币张数通过初始化命令设置，参见2.1 系统初始化。  
1.1规格说明  
1. 商品:每种商品包含商品名称、单价、数量三种属性，其中商品名不重复。考生不能修改商品名称和单价，初始化命令设置商品数量。这些信息在考试框架中进行定义，考生在实现功能代码时可直接使用。  
商品 名称  
单价  
数量  
A1 2 X  
A2 3 X  
A3 4 X  
A4 5 X  
A5 8 X  
A6 6 X  
2. 存钱盒信息：钱币面额、张数两种属性。初始化命令设置各种面额钱币张数。这些信息在考试框架中进行定义，考生在实现功能代码时可直接使用。  
钱币面额  
张数  
10元  
X  
5元  
X  
2元 X  
1元 X  
3. 退币原则 ：  
1) 根据系统存钱盒内钱币的 信息 ，按钱币总张数最少的原则进行退币。  
2) 如果因零钱不足导致不能退币，则尽最大可能退币，以减少用户损失。  
例如：假设存钱盒内只有4张2元，无其它面额钱币。如果需要退币7元，系统因零钱不足无法退币，则继续尝试退币6元，最终系统成功退币3张2元,用户损失1元钱币。  
4. 投币操作说明：每次投币成功，投入的钱币面额累加到投币余额；同时，本次投入的钱币放入存钱盒中，存钱盒相应面额钱币增加。  
5. 投币余额：指当前自动售货机中用户剩余的可购买商品的钱币总额；例如：投入2元面额的钱币，投币余额增加2元；购买一件价格2元的商品，投币余额减少2元；  
6. 退币操作说明：退币操作需要遵守 退币原则 ；退币成功后，投币余额清零，同时扣除存钱盒相应的金额。  
7. 购买商品操作说明：一次仅允许购买一件商品；购买商品成功后，自动售货机中对应商品数量减1，投币余额扣除本次购买商品的价格。  
2 操作说明  
命令字与第一个参数间使用一个空格分隔，多条命令采用分号隔开。考试系统会对输入命令格式进行处理，考生不需要关注输入命令格式的合法性，只需要实现命令处理函数。  
2.1 系统初始化  
命令格式：  
r A1 数量 -A2 数量 -A3 数量 -A4 数量 -A5 数量 -A6 数量 1 元张数 -2 元张数 -5 元张数 -10 元张数  
参数名称  
参数说明  
类型  
取值范围  
A1数量  
商品A1数量  
整数  
[0,30]  
A2数量  
商品A2数量  
整数  
[0,30]  
A3数量  
商品A3数量  
整数  
[0,30]  
A4数量  
商品A4数量  
整数  
[0,30]  
A5数量  
商品A5数量  
整数  
[0,30]  
A6数量  
商品A6数量  
整数  
[0,30]  
1元张数  
面额1元钱币张数  
整数  
[0,30]  
2元张数  
面额2元钱币张数  
整数  
[0,30]  
5元张数  
面额5元钱币张数  
整数  
[0,30]  
10元张数  
面额10元钱币张数  
整数  
[0,30]  
商品和各种面额钱币取值范围只是作为初始化命令的限制，其它场景下不限制取值范围；考试框架已经实现取值范围的检查，考生不需要关注。  
功能说明：设置自动售货机中商品数量和存钱盒各种面额的钱币张数；  
约束说明：系统在任意阶段均可执行r初始化系统；考生不需要关注参数的合法性，不需要关注增加或缺少参数的场景；  
输出说明：输出操作成功提示（执行完r命令后系统会自动输出操作结果，考生不需要再次调用输出函数），例：  
命令 输出 含义  
r 6-5-4-3-2-1 4-3-2-1; S001:Initialization is successful 初始化成功  
2.2 投币  
命令格式：p 钱币面额  
功能说明：  
（1） 如果投入非1元、2元、5元、10元的钱币面额（钱币面额不考虑负数、字符等非正整数的情况），输出“E002:Denomination error”；  
（2） 如果存钱盒中1元和2元面额钱币总额小于本次投入的钱币面额，输出“E003:Change is not enough, pay fail”，但投入1元和2元面额钱币不受此限制。  
（3） 如果自动售货机中商品全部销售完毕，投币失败。输出“E005:All the goods sold out”；  
（4） 如果投币成功，输出“S002:Pay success,balance=X”；  
约束说明：  
（1） 系统在任意阶段都可以投币；  
（2） 一次投币只能投一张钱币；  
（3） 同等条件下，错误码的优先级：E002 > E003 > E005；  
输出说明：如果投币成功，输出“S002:Pay success,balance=X”。  
例：  
命令  
输出  
p 10;  
S002:Pay success,balance=10  
2.3 购买商品  
命令格式：b 商品名称  
功能说明：  
（1） 如果购买的商品不在商品列表中，输出“E006:Goods does not exist”；  
（2） 如果所购买的商品的数量为0，输出“E007:The goods sold out”；  
（3） 如果投币余额小于待购买商品价格，输出“E008:Lack of balance”；  
（4） 如果购买成功，输出“S003:Buy success,balance=X”；  
约束说明：  
（1） 一次购买操作仅能购买一件商品，可以多次购买；  
（2） 同等条件下，错误码的优先级：E006 > E007 > E008；  
输出说明：  
如果购买成功，输出“S003:Buy success,balance=X”。  
例:  
命令  
输出  
b A1;  
S003:Buy success,balance=8  
2.4 退币  
命令格式：c  
功能说明：  
（1） 如果投币余额等于0的情况下，输出“E009:Work failure”；  
（2） 如果投币余额大于0的情况下，按照 退币原则 进行“找零”，输出退币信息；  
约束说明：  
（1） 系统在任意阶段都可以退币；  
（2） 退币方式必须按照 退币原则 进行退币；  
输出说明：如果退币成功，按照 退币原则 输出退币信息。  
例，退5元钱币:  
命令  
输出  
c;  
1 yuan coin number=0  
2 yuan coin number=0  
5 yuan coin number=1  
10 yuan coin number=0  
2.5 查询  
命令格式：q 查询类别  
功能说明：  
（1） 查询自动售货机中商品信息，包含商品名称、单价、数量。 根据商品数量从大到小进行排序；商品数量相同时，按照商品名称的先后顺序进行排序 。  
例如：A1的商品名称先于A2的商品名称，A2的商品名称先于A3的商品名称。  
（2） 查询存钱盒信息，包含各种面额钱币的张数；  
（3） 查询类别如下表所示:  
查询类别  
查询内容  
0  
查询商品信息  
1 查询存钱盒信息  
如果“查询类别”参数错误，输出“E010:Parameter error”。“查询类别”参数错误时，不进行下面的处理；  
输出说明：  
“查询类别”为0时，输出自动售货机中所有商品信息（商品名称单价数量）例：  
命令  
输出  
q 0;  
A1 2 6  
A2 3 5  
A3 4 4  
A4 5 3  
A5 8 2  
A6 6 0  
“查询类别”为1时，输出存钱盒信息（各种面额钱币的张数），格式固定。例：  
命令  
输出  
q 1;  
1 yuan coin number=4  
2 yuan coin number=3  
5 yuan coin number=2  
10 yuan coin number=1  
  
输入描述：  
依照说明中的命令码格式输入命令。  
  
输出描述：  
输出执行结果  
  
示例1  
输入：  
r 22-18-21-21-7-20 3-23-10-6;c;q0;p 1;b A6;c;b A5;b A1;c;q1;p 5;  
  
  
输出：  
S001:Initialization is successful  
E009:Work failure  
E010:Parameter error  
S002:Pay success,balance=1  
E008:Lack of balance  
1 yuan coin number=1  
2 yuan coin number=0  
5 yuan coin number=0  
10 yuan coin number=0  
E008:Lack of balance  
E008:Lack of balance  
E009:Work failure  
E010:Parameter error  
S002:Pay success,balance=5

描述  
首先输入要输入的整数个数n，然后输入n个整数。输出为n个整数中负数的个数，和所有正整数的平均值，结果保留一位小数。  
0即不是正整数，也不是负数，不计入计算。如果没有正数，则平均值为0。  
  
数据范围： 1 \le n\ \le 2000 \1≤n ≤2000 ，输入的整数都满足 |val| \le 1000 \∣val∣≤1000   
输入描述：  
首先输入一个正整数n，  
然后输入n个整数。  
  
输出描述：  
输出负数的个数，和所有正整数的平均值。  
  
示例1  
输入：  
11   
1 2 3 4 5 6 7 8 9 0 -1  
  
输出：  
1 5.0  
  
示例2  
输入：  
3  
0 0 0  
  
输出：  
0 0.0

描述  
将一个字符串中所有的整数前后加上符号“\*”，其他字符保持不变。连续的数字视为一个整数。  
  
  
数据范围：字符串长度满足 1 \le n \le 100 \1≤n≤100   
输入描述：  
输入一个字符串  
  
输出描述：  
字符中所有出现的数字前后加上符号“\*”，其他字符保持不变  
  
示例1  
输入：  
Jkdi234klowe90a3  
  
输出：  
Jkdi\*234\*klowe\*90\*a\*3\*

描述  
考试题目和要点：  
  
1、中文大写金额数字前应标明“人民币”字样。中文大写金额数字应用壹、贰、叁、肆、伍、陆、柒、捌、玖、拾、佰、仟、万、亿、元、角、分、零、整等字样填写。  
  
2、中文大写金额数字到“元”为止的，在“元”之后，应写“整字，如532.00应写成“人民币伍佰叁拾贰元整”。在”角“和”分“后面不写”整字。  
  
3、阿拉伯数字中间有“0”时，中文大写要写“零”字，阿拉伯数字中间连续有几个“0”时，中文大写金额中间只写一个“零”字，如6007.14，应写成“人民币陆仟零柒元壹角肆分“。  
4、10应写作“拾”，100应写作“壹佰”。例如，1010.00应写作“人民币壹仟零拾元整”，110.00应写作“人民币壹佰拾元整”  
5、十万以上的数字接千不用加“零”，例如，30105000.00应写作“人民币叁仟零拾万伍仟元整”  
  
输入描述：  
输入一个double数  
  
输出描述：  
输出人民币格式  
  
示例1  
输入：  
151121.15  
  
输出：  
人民币拾伍万壹仟壹佰贰拾壹元壹角伍分  
  
示例2  
输入：  
1010.00  
  
输出：  
人民币壹仟零拾元整

描述  
请实现一个计票统计系统。你会收到很多投票，其中有合法的也有不合法的，请统计每个候选人得票的数量以及不合法的票数。  
（注：不合法的投票指的是投票的名字不存在n个候选人的名字中！！）  
  
数据范围：每组输入中候选人数量满足 1 \le n \le 100 \1≤n≤100 ，总票数量满足 1 \le n \le 100 \1≤n≤100   
输入描述：  
第一行输入候选人的人数n，第二行输入n个候选人的名字（均为大写字母的字符串），第三行输入投票人的人数，第四行输入投票。  
  
输出描述：  
按照输入的顺序，每行输出候选人的名字和得票数量（以" : "隔开，注：英文冒号左右两边都有一个空格！），最后一行输出不合法的票数，格式为"Invalid : "+不合法的票数。  
  
示例1  
输入：  
4  
A B C D  
8  
A D E CF A GG A B  
  
输出：  
A : 3  
B : 1  
C : 0  
D : 1  
Invalid : 3  
  
说明：  
E CF GG三张票是无效的，所以Invalid的数量是3.

描述  
输入int型数组，询问该数组能否分成两组，使得两组中各元素加起来的和相等，并且，所有5的倍数必须在其中一个组中，所有3的倍数在另一个组中（不包括5的倍数），不是5的倍数也不是3的倍数能放在任意一组，可以将数组分为空数组，能满足以上条件，输出true；不满足时输出false。  
  
数据范围：每个数组大小满足 1 \le n \le 50 \1≤n≤50 ，输入的数据大小满足 |val| \le 500 \∣val∣≤500   
输入描述：  
第一行是数据个数，第二行是输入的数据  
  
输出描述：  
返回true或者false  
  
示例1  
输入：  
4  
1 5 -5 1  
  
输出：  
true  
  
说明：  
第一组：5 -5 1  
第二组：1   
示例2  
输入：  
3  
3 5 8  
  
输出：  
false  
  
说明：  
由于3和5不能放在同一组，所以不存在一种分法。

描述  
输入一个字符串，返回其最长的数字子串，以及其长度。若有多个最长的数字子串，则将它们全部输出（按原字符串的相对位置）  
本题含有多组样例输入。  
  
数据范围：字符串长度 1 \le n \le 200 \1≤n≤200 ， 保证每组输入都至少含有一个数字  
输入描述：  
输入一个字符串。1<=len(字符串)<=200  
  
输出描述：  
输出字符串中最长的数字字符串和它的长度，中间用逗号间隔。如果有相同长度的串，则要一块儿输出（中间不要输出空格）。  
  
示例1  
输入：  
abcd12345ed125ss123058789  
a8a72a6a5yy98y65ee1r2  
  
输出：  
123058789,9  
729865,2  
  
说明：  
样例一最长的数字子串为123058789，长度为9  
样例二最长的数字子串有72,98,65，长度都为2

描述  
请计算n\*m的棋盘格子（n为横向的格子数，m为竖向的格子数）从棋盘左上角出发沿着边缘线从左上角走到右下角，总共有多少种走法，要求不能走回头路，即：只能往右和往下走，不能往左和往上走。  
  
注：沿棋盘格之间的边缘线行走  
  
数据范围： 1 \le n,m \le 8 \1≤n,m≤8   
  
  
输入描述：  
输入两个正整数n和m，用空格隔开。(1≤n,m≤8)  
  
输出描述：  
输出一行结果  
  
示例1  
输入：  
2 2  
  
输出：  
6

描述  
IPV4地址可以用一个32位无符号整数来表示，一般用点分方式来显示，点将IP地址分成4个部分，每个部分为8位，表示成一个无符号整数（因此正号不需要出现），如10.137.17.1，是我们非常熟悉的IP地址，一个IP地址串中没有空格出现（因为要表示成一个32数字）。  
  
现在需要你用程序来判断IP是否合法。  
  
数据范围：数据组数：1\le t\le 18\1≤t≤18   
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
  
输入描述：  
输入一个ip地址，保证不包含空格  
  
输出描述：  
返回判断的结果YES or NO  
  
示例1  
输入：  
255.255.255.1000  
  
输出：  
NO

描述  
计算24点是一种扑克牌益智游戏，随机抽出4张扑克牌，通过加(+)，减(-)，乘(\*), 除(/)四种运算法则计算得到整数24，本问题中，扑克牌通过如下字符或者字符串表示，其中，小写joker表示小王，大写JOKER表示大王：  
  
3 4 5 6 7 8 9 10 J Q K A 2 joker JOKER  
  
本程序要求实现：输入4张牌，输出一个算式，算式的结果为24点。  
  
详细说明：  
  
1.运算只考虑加减乘除运算，没有阶乘等特殊运算符号，没有括号，友情提醒，整数除法要当心，是属于整除，比如2/3=0，3/2=1；  
2.牌面2~10对应的权值为2~10, J、Q、K、A权值分别为为11、12、13、1；  
3.输入4张牌为字符串形式，以一个空格隔开，首尾无空格；如果输入的4张牌中包含大小王，则输出字符串“ERROR”，表示无法运算；  
4.输出的算式格式为4张牌通过+-\*/四个运算符相连，中间无空格，4张牌出现顺序任意，只要结果正确；  
5.输出算式的运算顺序从左至右，不包含括号，如1+2+3\*4的结果为24，2 A 9 A不能变为(2+1)\*(9-1)=24  
6.如果存在多种算式都能计算得出24，只需输出一种即可，如果无法得出24，则输出“NONE”表示无解。  
7.因为都是扑克牌，不存在单个牌为0的情况，且没有括号运算，除数(即分母)的数字不可能为0  
  
数据范围：一行由4张牌组成的字符串  
输入描述：  
输入4张牌为字符串形式，以一个空格隔开，首尾无空格；  
  
输出描述：  
输出怎么运算得到24，如果无法得出24，则输出“NONE”表示无解，如果输入的4张牌中包含大小王，则输出字符串“ERROR”，表示无法运算；  
示例1  
输入：  
A A A A  
  
输出：  
NONE  
  
说明：  
不能实现   
示例2  
输入：  
4 2 K A  
  
输出：  
K-A\*4/2  
  
说明：  
 A+K\*2-4也是一种答案，输出任意一种即可   
示例3  
输入：  
B 5 joker 4  
  
输出：  
ERROR  
  
说明：  
 存在joker，输出ERROR   
示例4  
输入：  
K Q 6 K  
  
输出：  
NONE  
  
说明：  
按一般的计算规则来看，K+K-(Q/6)=24 或 K-((Q/6)-K)=24，但是因为这个题目的运算不许有括号，所以去掉括号后变为 K+K-Q/6=26-Q/6=14/6=2 或 K-Q/6-K=1/6-K=0-K=-13，其它情况也不能运算出24点，故不存在，输出NONE

描述  
扑克牌游戏大家应该都比较熟悉了，一副牌由54张组成，含3~A、2各4张，小王1张，大王1张。牌面从小到大用如下字符和字符串表示（其中，小写joker表示小王，大写JOKER表示大王）：  
3 4 5 6 7 8 9 10 J Q K A 2 joker JOKER  
输入两手牌，两手牌之间用"-"连接，每手牌的每张牌以空格分隔，"-"两边没有空格，如：4 4 4 4-joker JOKER。  
请比较两手牌大小，输出较大的牌，如果不存在比较关系则输出ERROR。  
基本规则：  
（1）输入每手牌可能是个子、对子、顺子（连续5张）、三个、炸弹（四个）和对王中的一种，不存在其他情况，由输入保证两手牌都是合法的，顺子已经从小到大排列；  
（2）除了炸弹和对王可以和所有牌比较之外，其他类型的牌只能跟相同类型的存在比较关系（如，对子跟对子比较，三个跟三个比较），不考虑拆牌情况（如：将对子拆分成个子）；  
（3）大小规则跟大家平时了解的常见规则相同，个子、对子、三个比较牌面大小；顺子比较最小牌大小；炸弹大于前面所有的牌，炸弹之间比较牌面大小；对王是最大的牌；  
  
（4）输入的两手牌不会出现相等的情况。  
  
数据范围：字符串长度：3\le s\le 10\3≤s≤10   
  
  
  
输入描述：  
输入两手牌，两手牌之间用"-"连接，每手牌的每张牌以空格分隔，"-"两边没有空格，如 4 4 4 4-joker JOKER。  
  
输出描述：  
输出两手牌中较大的那手，不含连接符，扑克牌顺序不变，仍以空格隔开；如果不存在比较关系则输出ERROR。  
  
示例1  
输入：  
4 4 4 4-joker JOKER  
  
输出：  
joker JOKER

描述  
密码按如下规则进行计分，并根据不同的得分为密码进行安全等级划分。  
  
一、密码长度:  
5 分: 小于等于4 个字符  
10 分: 5 到7 字符  
25 分: 大于等于8 个字符  
  
二、字母:  
0 分: 没有字母  
10 分: 密码里的字母全都是小（大）写字母  
20 分: 密码里的字母符合”大小写混合“  
  
三、数字:  
0 分: 没有数字  
10 分: 1 个数字  
20 分: 大于1 个数字  
  
四、符号:  
0 分: 没有符号  
10 分: 1 个符号  
25 分: 大于1 个符号  
  
五、奖励（只能选符合最多的那一种奖励）:  
2 分: 字母和数字  
3 分: 字母、数字和符号  
5 分: 大小写字母、数字和符号  
  
最后的评分标准:  
>= 90: 非常安全  
>= 80: 安全（Secure）  
>= 70: 非常强  
>= 60: 强（Strong）  
>= 50: 一般（Average）  
>= 25: 弱（Weak）  
>= 0: 非常弱（Very\_Weak）  
  
对应输出为：  
  
VERY\_SECURE  
SECURE  
VERY\_STRONG  
STRONG  
AVERAGE  
WEAK  
VERY\_WEAK  
  
请根据输入的密码字符串，进行安全评定。  
  
注：  
字母：a-z, A-Z  
数字：0-9  
符号包含如下： (ASCII码表可以在UltraEdit的菜单view->ASCII Table查看)  
!"#$%&'()\*+,-./ (ASCII码：0x21~0x2F)  
:;<=>?@ (ASCII码：0x3A~0x40)  
[\]^\_` (ASCII码：0x5B~0x60)  
{|}~ (ASCII码：0x7B~0x7E)  
  
提示:  
1 <= 字符串的长度<= 300  
输入描述：  
输入一个string的密码  
  
输出描述：  
输出密码等级  
  
示例1  
输入：  
38$@NoNoN  
  
输出：  
VERY\_SECURE  
  
说明：  
样例的密码长度大于等于8个字符，得25分；大小写字母都有所以得20分；有两个数字，所以得20分；包含大于1符号，所以得25分；由于该密码包含大小写字母、数字和符号，所以奖励部分得5分，经统计得该密码的密码强度为25+20+20+25+5=95分。  
   
示例2  
输入：  
Jl)M:+  
  
输出：  
AVERAGE  
  
说明：  
示例2的密码强度为10+20+0+25+0=55分。

描述  
求一个int类型数字对应的二进制数字中1的最大连续数，例如3的二进制为00000011，最大连续2个1  
  
数据范围：数据组数：1\le t\le 5\1≤t≤5 ，1\le n\le 500000\1≤n≤500000   
进阶：时间复杂度：O(logn)\O(logn) ，空间复杂度：O(1)\O(1)   
输入描述：  
输入一个int类型数字  
  
输出描述：  
输出转成二进制之后连续1的个数  
  
示例1  
输入：  
200  
  
输出：  
2  
  
说明：  
200的二进制表示是11001000，最多有2个连续的1。

描述  
给定一个仅包含小写字母的字符串，求它的最长回文子串的长度。  
所谓回文串，指左右对称的字符串。  
所谓子串，指一个字符串删掉其部分前缀和后缀（也可以不删）的字符串  
数据范围：字符串长度1\le s\le 350\1≤s≤350   
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
输入描述：  
输入一个仅包含小写字母的字符串  
  
输出描述：  
返回最长回文子串的长度  
  
示例1  
输入：  
cdabbacc  
  
输出：  
4  
  
说明：  
abba为最长的回文子串

描述  
找出给定字符串中大写字符(即'A'-'Z')的个数。  
数据范围：字符串长度：1\le |s|\le 250\1≤∣s∣≤250   
字符串中可能包含空格或其他字符  
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
输入描述：  
对于每组样例，输入一行，代表待统计的字符串  
  
输出描述：  
输出一个整数，代表字符串中大写字母的个数  
  
示例1  
输入：  
A 1 0 1 1150175017(&^%&$vabovbaoadd 123#$%#%#O  
  
输出：  
2

描述  
有一个m\*n\m∗n 大小的数据表，你会依次进行以下5种操作：  
1.输入m\m 和n\n ，初始化m\*n\m∗n 大小的表格。  
2.输入x\_1x   
1  
  
 、y\_1y   
1  
  
 、x\_2x   
2  
  
 、y\_2y   
2  
  
 ，交换坐标在(x\_1,y\_1)(x   
1  
  
 ,y   
1  
  
 )和(x\_2,y\_2)(x   
2  
  
 ,y   
2  
  
 )的两个数。  
3.输入x\x ，在第x\x 行上方添加一行。  
4.输入y\y ，在第y\y 列左边添加一列。  
5.输入x\x 、y\y ，查找坐标为(x,y)\(x,y) 的单元格的值。  
请编写程序，判断对表格的各种操作是否合法。  
  
详细要求:  
  
1.数据表的最大规格为9行\*9列，对表格进行操作时遇到超出规格应该返回错误。  
2.对于插入操作，如果插入后行数或列数超过9了则应返回错误。如果插入成功了则将数据表恢复至初始化的m\*n\m∗n 大小，多出的数据则应舍弃。  
3.所有输入坐标操作，对m\*n\m∗n 大小的表格，行号坐标只允许0~m-1，列号坐标只允许0~n-1。超出范围应该返回错误。  
  
本题含有多组样例输入！行列从0开始标号  
数据范围：数据组数：1\le t\le 5\1≤t≤5   
进阶：时间复杂度：O(1)\O(1) ，空间复杂度：O(1)\O(1)   
输入描述：  
输入数据按下列顺序输入：  
1 表格的行列值  
2 要交换的两个单元格的行列值  
3 输入要插入的行的数值  
4 输入要插入的列的数值  
5 输入要查询的单元格的坐标  
  
输出描述：  
输出按下列顺序输出：  
1 初始化表格是否成功，若成功则返回0， 否则返回-1  
2 输出交换单元格是否成功  
3 输出插入行是否成功  
4 输出插入列是否成功  
5 输出查询单元格数据是否成功  
  
示例1  
输入：  
4 9  
5 1 2 6  
0  
8  
2 3  
4 7  
4 2 3 2  
3  
3  
4 7  
  
输出：  
0  
-1  
0  
-1  
0  
0  
-1  
0  
0  
-1  
  
说明：  
本组样例共有2组样例输入。  
第一组样例：  
1.初始化数据表为4行9列，成功  
2.交换第5行1列和第2行6列的数据，失败。因为行的范围应该是(0,3)，不存在第5行。  
3.在第0行上方添加一行，成功。  
4.在第8列左边添加一列，失败。因为列的总数已经达到了9的上限。  
5.查询第2行第3列的值，成功。  
第二组样例：  
1.初始化数据表为4行7列，成功  
2.交换第4行2列和第3行2列的数据，失败。因为行的范围应该是(0,3)，不存在第4行。  
3.在第3行上方添加一行，成功。  
4.在第3列左边添加一列，成功。  
5.查询第4行7列的值，失败。因为虽然添加了一行一列，但数据表会在添加后恢复成4行7列的形态，所以行的区间仍然在[0,3]，列的区间仍然在[0,6]，无法查询到(4,7)坐标。

描述  
分子为1的分数称为埃及分数。现输入一个真分数(分子比分母小的分数，叫做真分数)，请将该分数分解为埃及分数。如：8/11 = 1/2+1/5+1/55+1/110。  
注：真分数指分子小于分母的分数，分子和分母有可能gcd不为1！  
如有多个解，请输出任意一个。  
  
  
输入描述：  
输入一个真分数，String型  
  
输出描述：  
输出分解后的string  
  
示例1  
输入：  
8/11  
2/4  
  
输出：  
1/2+1/5+1/55+1/110  
1/3+1/6  
  
说明：  
第二个样例直接输出1/2也是可以的

描述  
判断短字符串S中的所有字符是否在长字符串T中全部出现。  
请注意本题有多组样例输入。  
数据范围:1\le len(S),len(T)\le200\1≤len(S),len(T)≤200   
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
输入描述：  
输入两个字符串。第一个为短字符串，第二个为长字符串。两个字符串均由小写字母组成。  
  
输出描述：  
如果短字符串的所有字符均在长字符串中出现过，则输出字符串"true"。否则输出字符串"false"。  
  
示例1  
输入：  
bc  
abc  
  
输出：  
true  
  
说明：  
其中abc含有bc，输出"true"

描述  
题目标题：  
  
将两个整型数组按照升序合并，并且过滤掉重复数组元素。  
输出时相邻两数之间没有空格。  
  
  
  
输入描述：  
输入说明，按下列顺序输入：  
1 输入第一个数组的个数  
2 输入第一个数组的数值  
3 输入第二个数组的个数  
4 输入第二个数组的数值  
  
输出描述：  
输出合并之后的数组  
  
示例1  
输入：  
3  
1 2 5  
4  
-1 0 3 2  
  
输出：  
-101235

描述  
给定一个正整数N代表火车数量，0<N<10，接下来输入火车入站的序列，一共N辆火车，每辆火车以数字1-9编号，火车站只有一个方向进出，同时停靠在火车站的列车中，只有后进站的出站了，先进站的才能出站。  
要求输出所有火车出站的方案，以字典序排序输出。  
数据范围：1\le n\le 10\1≤n≤10   
进阶：时间复杂度：O(n!)\O(n!) ，空间复杂度：O(n)\O(n)   
输入描述：  
第一行输入一个正整数N（0 < N <= 10），第二行包括N个正整数，范围为1到10。  
  
输出描述：  
输出以字典序从小到大排序的火车出站序列号，每个编号以空格隔开，每个输出序列换行，具体见sample。  
  
示例1  
输入：  
3  
1 2 3  
  
输出：  
1 2 3  
1 3 2  
2 1 3  
2 3 1  
3 2 1  
  
说明：  
第一种方案：1进、1出、2进、2出、3进、3出  
第二种方案：1进、1出、2进、3进、3出、2出  
第三种方案：1进、2进、2出、1出、3进、3出  
第四种方案：1进、2进、2出、3进、3出、1出  
第五种方案：1进、2进、3进、3出、2出、1出  
请注意，[3,1,2]这个序列是不可能实现的。

描述  
验证尼科彻斯定理，即：任何一个整数m的立方都可以写成m个连续奇数之和。  
  
例如：  
  
1^3=1  
  
2^3=3+5  
  
3^3=7+9+11  
  
4^3=13+15+17+19  
  
输入一个正整数m（m≤100），将m的立方写成m个连续奇数之和的形式输出。  
数据范围：1\le m\le 100\1≤m≤100   
进阶：时间复杂度：O(m)\O(m) ，空间复杂度：O(1)\O(1)   
  
输入描述：  
输入一个int整数  
  
输出描述：  
输出分解后的string  
  
示例1  
输入：  
6  
  
输出：  
31+33+35+37+39+41

描述  
给定两个只包含小写字母的字符串，计算两个字符串的最大公共子串的长度。  
  
注：子串的定义指一个字符串删掉其部分前缀和后缀（也可以不删）后形成的字符串。  
数据范围：字符串长度：1\le s\le 150\1≤s≤150   
进阶：时间复杂度：O(n^3)\O(n   
3  
 ) ，空间复杂度：O(n)\O(n)   
输入描述：  
输入两个只包含小写字母的字符串  
  
输出描述：  
输出一个整数，代表最大公共子串的长度  
  
示例1  
输入：  
asdfas  
werasdfaswer  
  
输出：  
6

描述  
在命令行输入如下命令：  
  
xcopy /s c:\\ d:\\e，  
  
各个参数如下：  
  
参数1：命令字xcopy  
  
参数2：字符串/s  
  
参数3：字符串c:\\  
  
参数4: 字符串d:\\e  
  
请编写一个参数解析程序，实现将命令行各个参数解析出来。  
  
  
解析规则：  
  
1.参数分隔符为空格  
2.对于用""包含起来的参数，如果中间有空格，不能解析为多个参数。比如在命令行输入xcopy /s "C:\\program files" "d:\"时，参数仍然是4个，第3个参数应该是字符串C:\\program files，而不是C:\\program，注意输出参数时，需要将""去掉，引号不存在嵌套情况。  
3.参数不定长  
  
4.输入由用例保证，不会出现不符合要求的输入  
数据范围：字符串长度：1\le s\le 1000\1≤s≤1000   
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
输入描述：  
输入一行字符串，可以有空格  
  
输出描述：  
输出参数个数，分解后的参数，每个参数都独占一行  
  
示例1  
输入：  
xcopy /s c:\\ d:\\e  
  
输出：  
4  
xcopy  
/s  
c:\\  
d:\\e

描述  
根据输入的日期，计算是这一年的第几天。  
保证年份为4位数且日期合法。  
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(1)\O(1)   
输入描述：  
输入一行，每行空格分割，分别是年，月，日  
  
输出描述：  
输出是这一年的第几天  
示例1  
输入：  
2012 12 31  
  
输出：  
366  
  
示例2  
输入：  
1982 3 4  
  
输出：  
63

描述  
公元五世纪，我国古代数学家张丘建在《算经》一书中提出了“百鸡问题”：鸡翁一值钱五，鸡母一值钱三，鸡雏三值钱一。百钱买百鸡，问鸡翁、鸡母、鸡雏各几何？  
现要求你打印出所有花一百元买一百只鸡的方式。  
输入描述：  
输入任何一个整数，即可运行程序。  
  
输出描述：  
 输出有数行，每行三个整数，分别代表鸡翁，母鸡，鸡雏的数量  
  
示例1  
输入：  
1  
  
输出：  
0 25 75  
4 18 78  
8 11 81  
12 4 84

描述  
问题描述：在计算机中，通配符一种特殊语法，广泛应用于文件搜索、数据库、正则表达式等领域。现要求各位实现字符串通配符的算法。  
要求：  
实现如下2个通配符：  
\*：匹配0个或以上的字符（注：能被\*和?匹配的字符仅由英文字母和数字0到9组成，下同）  
？：匹配1个字符  
  
注意：匹配时不区分大小写。  
  
输入：  
通配符表达式；  
一组字符串。  
输出：  
  
返回不区分大小写的匹配结果，匹配成功输出true，匹配失败输出false  
数据范围：字符串长度：1\le s\le 100\1≤s≤100   
进阶：时间复杂度：O(n^2)\O(n   
2  
 ) ，空间复杂度：O(n)\O(n)   
输入描述：  
先输入一个带有通配符的字符串，再输入一个需要匹配的字符串  
  
输出描述：  
返回不区分大小写的匹配结果，匹配成功输出true，匹配失败输出false  
  
示例1  
输入：  
te?t\*.\*  
txt12.xls  
  
输出：  
false  
  
示例2  
输入：  
z  
zz  
  
输出：  
false  
  
示例3  
输入：  
pq  
pppq  
  
输出：  
false  
  
示例4  
输入：  
\*\*Z  
0QZz  
  
输出：  
true  
  
示例5  
输入：  
?\*Bc\*?  
abcd  
  
输出：  
true  
  
示例6  
输入：  
h\*?\*a  
h#a  
  
输出：  
false  
  
说明：  
根据题目描述可知能被\*和?匹配的字符仅由英文字母和数字0到9组成，所以?不能匹配#，故输出false   
示例7  
输入：  
p\*p\*qp\*\*pq\*p\*\*p\*\*\*ppq  
pppppppqppqqppqppppqqqppqppqpqqqppqpqpppqpppqpqqqpqqp  
  
输出：  
false

描述  
矩阵乘法的运算量与矩阵乘法的顺序强相关。  
例如：  
  
A是一个50×10的矩阵，B是10×20的矩阵，C是20×5的矩阵  
  
计算A\*B\*C有两种顺序：((AB)C)或者(A(BC))，前者需要计算15000次乘法，后者只需要3500次。  
  
编写程序计算不同的计算顺序需要进行的乘法次数。  
  
数据范围：矩阵个数：1\le n\le 15 \1≤n≤15 ，行列数：1\le row\_i,col\_i\le 100\1≤row   
i  
  
 ,col   
i  
  
 ≤100 ，保证给出的字符串表示的计算顺序唯一。  
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
  
输入描述：  
输入多行，先输入要计算乘法的矩阵个数n，每个矩阵的行数，列数，总共2n的数，最后输入要计算的法则  
计算的法则为一个字符串，仅由左右括号和大写字母（'A'~'Z'）组成，保证括号是匹配的且输入合法！  
  
输出描述：  
输出需要进行的乘法次数  
  
示例1  
输入：  
3  
50 10  
10 20  
20 5  
(A(BC))  
  
输出：  
3500

描述  
给定一些同学的信息（名字，成绩）序列，请你将他们的信息按照成绩从高到低或从低到高的排列,相同成绩  
  
都按先录入排列在前的规则处理。  
  
例示：  
jack 70  
peter 96  
Tom 70  
smith 67  
  
从高到低 成绩  
peter 96  
jack 70  
Tom 70  
smith 67  
  
从低到高  
  
smith 67  
  
jack 70  
  
Tom 70  
peter 96  
  
注：0代表从高到低，1代表从低到高  
  
数据范围：人数：1\le n \le 200\1≤n≤200   
进阶：时间复杂度：O(nlogn)\O(nlogn) ，空间复杂度：O(n)\O(n)   
输入描述：  
第一行输入要排序的人的个数n，第二行输入一个整数表示排序的方式，之后n行分别输入他们的名字和成绩，以一个空格隔开  
  
输出描述：  
按照指定方式输出名字和成绩，名字和成绩之间以一个空格隔开  
  
示例1  
输入：  
3  
0  
fang 90  
yang 50  
ning 70  
  
输出：  
fang 90  
ning 70  
yang 50  
  
示例2  
输入：  
3  
1  
fang 90  
yang 50  
ning 70  
  
输出：  
yang 50  
ning 70  
fang 90

描述  
给出4个1-10的数字，通过加减乘除运算，得到数字为24就算胜利,除法指实数除法运算,运算符仅允许出现在两个数字之间,本题对数字选取顺序无要求，但每个数字仅允许使用一次，且需考虑括号运算  
此题允许数字重复，如3 3 4 4为合法输入，此输入一共有两个3，但是每个数字只允许使用一次，则运算过程中两个3都被选取并进行对应的计算操作。  
输入描述：  
读入4个[1,10]的整数，数字允许重复，测试用例保证无异常数字。  
  
输出描述：  
对于每组案例，输出一行表示能否得到24点，能输出true，不能输出false  
  
示例1  
输入：  
7 2 1 10  
  
输出：  
true

描述  
有6条配置命令，它们执行的结果分别是：  
  
命 令 执 行  
reset reset what  
reset board board fault  
board add where to add  
board delete no board at all  
reboot backplane impossible  
backplane abort install first  
he he unknown command  
注意：he he不是命令。  
  
为了简化输入，方便用户，以“最短唯一匹配原则”匹配（注：需从首字母开始进行匹配）：  
  
1、若只输入一字串，则只匹配一个关键字的命令行。例如输入：r，根据该规则，匹配命令reset，执行结果为：reset what；输入：res，根据该规则，匹配命令reset，执行结果为：reset what；  
2、若只输入一字串，但匹配命令有两个关键字，则匹配失败。例如输入：reb，可以找到命令reboot backpalne，但是该命令有两个关键词，所有匹配失败，执行结果为：unknown command  
  
3、若输入两字串，则先匹配第一关键字，如果有匹配，继续匹配第二关键字，如果仍不唯一，匹配失败。  
例如输入：r b，找到匹配命令reset board 和 reboot backplane，执行结果为：unknown command。  
例如输入：b a，无法确定是命令board add还是backplane abort，匹配失败。  
4、若输入两字串，则先匹配第一关键字，如果有匹配，继续匹配第二关键字，如果唯一，匹配成功。例如输入：bo a，确定是命令board add，匹配成功。  
5、若输入两字串，第一关键字匹配成功，则匹配第二关键字，若无匹配，失败。例如输入：b addr，无法匹配到相应的命令，所以执行结果为：unknow command。  
6、若匹配失败，打印“unknown command”  
  
注意：有多组输入。  
数据范围：数据组数：1\le t\le 800\1≤t≤800 ，字符串长度1\le s\le 20\1≤s≤20   
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
输入描述：  
多行字符串，每行字符串一条命令  
  
输出描述：  
执行结果，每条命令输出一行  
  
示例1  
输入：  
reset  
reset board  
board add  
board delet  
reboot backplane  
backplane abort  
  
输出：  
reset what  
board fault  
where to add  
no board at all  
impossible  
install first

描述  
查找两个字符串a,b中的最长公共子串。若有多个，输出在较短串中最先出现的那个。  
注：子串的定义：将一个字符串删去前缀和后缀（也可以不删）形成的字符串。请和“子序列”的概念分开！  
  
数据范围：字符串长度1\le length \le300 \1≤length≤300   
进阶：时间复杂度：O(n^3)\O(n   
3  
 ) ，空间复杂度：O(n)\O(n)   
输入描述：  
输入两个字符串  
  
输出描述：  
返回重复出现的字符  
示例1  
输入：  
abcdefghijklmnop  
abcsafjklmnopqrstuvw  
  
输出：  
jklmnop

描述  
MP3 Player因为屏幕较小，显示歌曲列表的时候每屏只能显示几首歌曲，用户要通过上下键才能浏览所有的歌曲。为了简化处理，假设每屏只能显示4首歌曲，光标初始的位置为第1首歌。  
  
  
现在要实现通过上下键控制光标移动来浏览歌曲列表，控制逻辑如下：  
  
歌曲总数<=4的时候，不需要翻页，只是挪动光标位置。  
  
光标在第一首歌曲上时，按Up键光标挪到最后一首歌曲；光标在最后一首歌曲时，按Down键光标挪到第一首歌曲。  
  
  
  
其他情况下用户按Up键，光标挪到上一首歌曲；用户按Down键，光标挪到下一首歌曲。  
  
  
  
2. 歌曲总数大于4的时候（以一共有10首歌为例）：  
  
  
特殊翻页：屏幕显示的是第一页（即显示第1 – 4首）时，光标在第一首歌曲上，用户按Up键后，屏幕要显示最后一页（即显示第7-10首歌），同时光标放到最后一首歌上。同样的，屏幕显示最后一页时，光标在最后一首歌曲上，用户按Down键，屏幕要显示第一页，光标挪到第一首歌上。  
  
  
  
一般翻页：屏幕显示的不是第一页时，光标在当前屏幕显示的第一首歌曲时，用户按Up键后，屏幕从当前歌曲的上一首开始显示，光标也挪到上一首歌曲。光标当前屏幕的最后一首歌时的Down键处理也类似。  
  
  
  
其他情况，不用翻页，只是挪动光标就行。  
  
数据范围：命令长度1\le s\le 100\1≤s≤100 ，歌曲数量1\le n \le 150\1≤n≤150   
进阶：时间复杂度：O(n)\O(n) ，空间复杂度：O(n)\O(n)   
输入描述：  
输入说明：  
1 输入歌曲数量  
2 输入命令 U或者D  
  
输出描述：  
输出说明  
1 输出当前列表  
2 输出当前选中歌曲  
  
示例1  
输入：  
10  
UUUU  
  
输出：  
7 8 9 10  
7

描述  
一个 DNA 序列由 A/C/G/T 四个字母的排列组合组成。 G 和 C 的比例（定义为 GC-Ratio ）是序列中 G 和 C 两个字母的总的出现次数除以总的字母数目（也就是序列长度）。在基因工程中，这个比例非常重要。因为高的 GC-Ratio 可能是基因的起始点。  
  
给定一个很长的 DNA 序列，以及限定的子串长度 N ，请帮助研究人员在给出的 DNA 序列中从左往右找出 GC-Ratio 最高且长度为 N 的第一个子串。  
DNA序列为 ACGT 的子串有: ACG , CG , CGT 等等，但是没有 AGT ， CT 等等  
  
数据范围：字符串长度满足 1 \le n \le 1000 \1≤n≤1000 ，输入的字符串只包含 A/C/G/T 字母  
输入描述：  
输入一个string型基因序列，和int型子串的长度  
  
输出描述：  
找出GC比例最高的子串,如果有多个则输出第一个的子串  
  
示例1  
输入：  
ACGT  
2  
  
输出：  
CG  
  
说明：  
ACGT长度为2的子串有AC,CG,GT3个，其中AC和GT2个的GC-Ratio都为0.5，CG为1，故输出CG   
示例2  
输入：  
AACTGTGCACGACCTGA  
5  
  
输出：  
GCACG  
  
说明：  
虽然CGACC的GC-Ratio也是最高，但它是从左往右找到的GC-Ratio最高的第2个子串，所以只能输出GCACG。

描述  
输入一个正整数，计算它在二进制下的1的个数。  
注意多组输入输出！！！！！！  
  
数据范围： 1 \le n \le 2^{31}-1 \1≤n≤2   
31  
 −1   
输入描述：  
输入一个整数  
  
输出描述：  
计算整数二进制中1的个数  
  
示例1  
输入：  
5  
  
输出：  
2  
  
说明：  
5的二进制表示是101，有2个1   
示例2  
输入：  
0  
  
输出：  
0

描述  
把m个同样的苹果放在n个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的分法？  
注意：如果有7个苹果和3个盘子，（5，1，1）和（1，5，1）被视为是同一种分法。  
  
数据范围：0 \le m \le 10 \0≤m≤10 ，1 \le n \le 10 \1≤n≤10 。  
  
输入描述：  
输入两个int整数  
  
输出描述：  
输出结果，int型  
  
示例1  
输入：  
7 3  
  
输出：  
8

描述  
任意一个偶数（大于2）都可以由2个素数组成，组成偶数的2个素数有很多种情况，本题目要求输出组成指定偶数的两个素数差值最小的素数对。  
  
数据范围：输入的数据满足 4 \le n \le 1000 \4≤n≤1000   
输入描述：  
输入一个大于2的偶数  
  
输出描述：  
从小到大输出两个素数  
  
示例1  
输入：  
20  
  
输出：  
7  
13  
  
示例2  
输入：  
4  
  
输出：  
2  
2

描述  
找出字符串中第一个只出现一次的字符  
  
  
数据范围：输入的字符串长度满足 1 \le n \le 1000 \1≤n≤1000   
  
  
输入描述：  
输入一个非空字符串  
  
输出描述：  
输出第一个只出现一次的字符，如果不存在输出-1  
  
示例1  
输入：  
asdfasdfo  
  
  
输出：  
o

描述  
输入n个整数，找出其中最小的k个整数并按升序输出  
  
本题有多组输入样例  
  
数据范围：1 \le n \le 1000 \1≤n≤1000 ，输入的整数满足 1 \le val \le 10000 \1≤val≤10000   
输入描述：  
第一行输入两个整数n和k  
第二行输入一个整数数组  
  
输出描述：  
从小到大输出最小的k个整数，用空格分开。  
  
示例1  
输入：  
5 2  
1 3 5 7 2  
  
输出：  
1 2

描述  
输入两个用字符串 str 表示的整数，求它们所表示的数之和。  
  
数据范围： 1 \le len(str) \le 10000 \1≤len(str)≤10000   
输入描述：  
输入两个字符串。保证字符串只含有'0'~'9'字符  
  
输出描述：  
输出求和后的结果  
  
示例1  
输入：  
9876543210  
1234567890  
  
输出：  
11111111100

描述  
完全数（Perfect number），又称完美数或完备数，是一些特殊的自然数。  
  
它所有的真因子（即除了自身以外的约数）的和（即因子函数），恰好等于它本身。  
  
例如：28，它有约数1、2、4、7、14、28，除去它本身28外，其余5个数相加，1+2+4+7+14=28。  
  
输入n，请输出n以内(含n)完全数的个数。  
  
数据范围： 1 \le n \le 5 \times 10^{5} \1≤n≤5×10   
5  
   
输入描述：  
输入一个数字n  
  
输出描述：  
输出不超过n的完全数的个数  
  
示例1  
输入：  
1000  
  
输出：  
3

描述  
输出 1到n之间 的与 7 有关数字的个数。  
一个数与7有关是指这个数是 7 的倍数，或者是包含 7 的数字（如 17 ，27 ，37 ... 70 ，71 ，72 ，73...）  
  
数据范围： 1 \le n \le 30000 \1≤n≤30000   
输入描述：  
一个正整数 n 。( n 不大于 30000 )  
  
输出描述：  
一个整数，表示1到n之间的与7有关的数字个数。  
  
示例1  
输入：  
20  
  
输出：  
3  
  
说明：  
输入20，1到20之间有关的数字包括7,14,17共3个。

描述  
给定一个字符串描述的算术表达式，计算出结果值。  
  
输入字符串长度不超过 100 ，合法的字符包括 ”+, -, \*, /, (, )” ， ”0-9” 。  
  
数据范围：运算过程中和最终结果均满足 |val| \le 2^{31}-1 \∣val∣≤2   
31  
 −1 ，即只进行整型运算，确保输入的表达式合法  
输入描述：  
输入算术表达式  
  
输出描述：  
计算出结果值  
  
示例1  
输入：  
400+5  
  
输出：  
405

描述  
  
  
以上三角形的数阵，第一行只有一个数1，以下每行的每个数，是恰好是它上面的数、左上角数和右上角的数，3个数之和（如果不存在某个数，认为该数就是0）。  
  
求第n行第一个偶数出现的位置。如果没有偶数，则输出-1。例如输入3,则输出2，输入4则输出3，输入2则输出-1。  
  
数据范围： 1 \le n \le 10^9 \1≤n≤10   
9  
   
  
输入描述：  
输入一个int整数  
  
输出描述：  
输出返回的int值  
  
示例1  
输入：  
4  
  
输出：  
3

描述  
Levenshtein 距离，又称编辑距离，指的是两个字符串之间，由一个转换成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。编辑距离的算法是首先由俄国科学家 Levenshtein 提出的，故又叫 Levenshtein Distance 。  
  
例如：  
  
字符串A: abcdefg  
  
字符串B: abcdef  
  
通过增加或是删掉字符 ”g” 的方式达到目的。这两种方案都需要一次操作。把这个操作所需要的次数定义为两个字符串的距离。  
  
要求：  
  
给定任意两个字符串，写出一个算法计算它们的编辑距离。  
  
  
数据范围：给定的字符串长度满足 1 \le len(str) \le 1000 \1≤len(str)≤1000   
  
  
输入描述：  
每组用例一共2行，为输入的两个字符串  
  
输出描述：  
每组用例输出一行，代表字符串的距离  
  
示例1  
输入：  
abcdefg  
abcdef  
  
输出：  
1

描述  
输入一个单向链表，输出该链表中倒数第k个结点，链表的倒数第1个结点为链表的尾指针。  
  
链表结点定义如下：  
struct ListNode  
{  
 int m\_nKey;  
 ListNode\* m\_pNext;  
};  
正常返回倒数第k个结点指针，异常返回空指针.  
要求：  
(1)正序构建链表;  
(2)构建后要忘记链表长度。  
数据范围：链表长度满足 1 \le n \le 1000 \1≤n≤1000 ， k \le n \k≤n ，链表中数据满足 0 \le val \le 10000 \0≤val≤10000   
  
本题有多组样例输入。  
  
  
输入描述：  
输入说明  
1 输入链表结点个数  
2 输入链表的值  
3 输入k的值  
  
输出描述：  
输出一个整数  
  
示例1  
输入：  
8  
1 2 3 4 5 6 7 8  
4  
  
输出：  
5

描述  
输入一个表达式（用字符串表示），求这个表达式的值。  
保证字符串中的有效字符包括[‘0’-‘9’],‘+’,‘-’, ‘\*’,‘/’ ,‘(’， ‘)’,‘[’, ‘]’,‘{’ ,‘}’。且表达式一定合法。  
  
数据范围：表达式计算结果和过程中满足 |val| \le 1000 \∣val∣≤1000 ，字符串长度满足 1 \le n \le 1000 \1≤n≤1000   
  
输入描述：  
输入一个算术表达式  
  
输出描述：  
得到计算结果  
  
示例1  
输入：  
3+2\*{1+2\*[-4/(8-6)+7]}  
  
输出：  
25

描述  
输入一个单向链表和一个节点的值，从单向链表中删除等于该值的节点，删除后如果链表中无节点则返回空指针。  
  
链表的值不能重复。  
  
构造过程，例如输入一行数据为:  
6 2 1 2 3 2 5 1 4 5 7 2 2  
则第一个参数6表示输入总共6个节点，第二个参数2表示头节点值为2，剩下的2个一组表示第2个节点值后面插入第1个节点值，为以下表示:  
1 2 表示为  
2->1  
链表为2->1  
  
3 2表示为  
2->3  
链表为2->3->1  
  
5 1表示为  
1->5  
链表为2->3->1->5  
  
4 5表示为  
5->4  
链表为2->3->1->5->4  
  
7 2表示为  
2->7  
链表为2->7->3->1->5->4  
  
最后的链表的顺序为 2 7 3 1 5 4  
  
最后一个参数为2，表示要删掉节点为2的值  
删除 结点 2  
  
则结果为 7 3 1 5 4  
  
数据范围：链表长度满足 1 \le n \le 1000 \1≤n≤1000 ，节点中的值满足 0 \le val \le 10000 \0≤val≤10000   
  
测试用例保证输入合法  
  
  
输入描述：  
输入一行，有以下4个部分：  
1 输入链表结点个数  
2 输入头结点的值  
3 按照格式插入各个结点  
4 输入要删除的结点的值  
  
输出描述：  
输出一行  
输出删除结点后的序列，每个数后都要加空格  
  
示例1  
输入：  
5 2 3 2 4 3 5 2 1 4 3  
  
输出：  
2 5 4 1  
  
说明：  
形成的链表为2->5->3->4->1  
删掉节点3，返回的就是2->5->4->1   
示例2  
输入：  
6 2 1 2 3 2 5 1 4 5 7 2 2  
  
输出：  
7 3 1 5 4  
  
说明：  
如题

描述  
输入一个字符串和一个整数 k ，截取字符串的前k个字符并输出  
  
数据范围：字符串长度满足 1 \le n \le 1000 \1≤n≤1000 ， 1 \le k \le n \1≤k≤n   
输入描述：  
1.输入待截取的字符串  
  
2.输入一个正整数k，代表截取的长度  
  
输出描述：  
截取后的字符串  
  
示例1  
输入：  
abABCcDEF  
6  
  
输出：  
abABCc  
  
示例2  
输入：  
bdxPKBhih  
6  
  
输出：  
bdxPKB

描述  
给出一个字符串，该字符串仅由小写字母组成，定义这个字符串的“漂亮度”是其所有字母“漂亮度”的总和。  
每个字母都有一个“漂亮度”，范围在1到26之间。没有任何两个不同字母拥有相同的“漂亮度”。字母忽略大小写。  
  
给出多个字符串，计算每个字符串最大可能的“漂亮度”。  
  
本题含有多组数据。  
  
数据范围：输入的名字长度满足 1 \le n \le 10000 \1≤n≤10000   
  
输入描述：  
第一行一个整数N，接下来N行每行一个字符串  
  
输出描述：  
每个字符串可能的最大漂亮程度  
  
示例1  
输入：  
2  
zhangsan  
lisi  
  
输出：  
192  
101  
  
说明：  
对于样例lisi，让i的漂亮度为26，l的漂亮度为25，s的漂亮度为24，lisi的漂亮度为25+26+24+26=101.

描述  
问题描述：数独（Sudoku）是一款大众喜爱的数字逻辑游戏。玩家需要根据9X9盘面上的已知数字，推算出所有剩余空格的数字，并且满足每一行、每一列、每一个3X3粗线宫内的数字均含1-9，并且不重复。  
例如：  
输入  
  
输出  
  
  
数据范围：输入一个 9\*9 的矩阵  
输入描述：  
包含已知数字的9X9盘面数组[空缺位以数字0表示]  
  
输出描述：  
完整的9X9盘面数组  
  
示例1  
输入：  
0 9 2 4 8 1 7 6 3  
4 1 3 7 6 2 9 8 5  
8 6 7 3 5 9 4 1 2  
6 2 4 1 9 5 3 7 8  
7 5 9 8 4 3 1 2 6  
1 3 8 6 2 7 5 9 4  
2 7 1 5 3 8 6 4 9  
3 8 6 9 1 4 2 5 7  
0 4 5 2 7 6 8 3 1  
  
输出：  
5 9 2 4 8 1 7 6 3  
4 1 3 7 6 2 9 8 5  
8 6 7 3 5 9 4 1 2  
6 2 4 1 9 5 3 7 8  
7 5 9 8 4 3 1 2 6  
1 3 8 6 2 7 5 9 4  
2 7 1 5 3 8 6 4 9  
3 8 6 9 1 4 2 5 7  
9 4 5 2 7 6 8 3 1

描述  
定义一个二维数组 N\*M ，如 5 × 5 数组下所示：  
  
  
int maze[5][5] = {  
0, 1, 0, 0, 0,  
0, 1, 1, 1, 0,  
0, 0, 0, 0, 0,  
0, 1, 1, 1, 0,  
0, 0, 0, 1, 0,  
};  
  
  
它表示一个迷宫，其中的1表示墙壁，0表示可以走的路，只能横着走或竖着走，不能斜着走，要求编程序找出从左上角到右下角的路线。入口点为[0,0],既第一格是可以走的路。  
  
  
数据范围： 2 \le n,m \le 10 \2≤n,m≤10 ， 输入的内容只包含 0 \le val \le 1 \0≤val≤1   
  
输入描述：  
输入两个整数，分别表示二维数组的行数，列数。再输入相应的数组，其中的1表示墙壁，0表示可以走的路。数据保证有唯一解,不考虑有多解的情况，即迷宫只有一条通道。  
  
输出描述：  
左上角到右下角的最短路径，格式如样例所示。  
  
示例1  
输入：  
5 5  
0 1 0 0 0  
0 1 1 1 0  
0 0 0 0 0  
0 1 1 1 0  
0 0 0 1 0  
  
输出：  
(0,0)  
(1,0)  
(2,0)  
(2,1)  
(2,2)  
(2,3)  
(2,4)  
(3,4)  
(4,4)  
  
示例2  
输入：  
5 5  
0 1 0 0 0  
0 1 0 1 0  
0 0 0 0 1  
0 1 1 1 0  
0 0 0 0 0  
  
输出：  
(0,0)  
(1,0)  
(2,0)  
(3,0)  
(4,0)  
(4,1)  
(4,2)  
(4,3)  
(4,4)  
  
  
说明：  
注意：不能斜着走！！

描述  
Jessi初学英语，为了快速读出一串数字，编写程序将数字转换成英文：  
  
具体规则如下:  
1.在英语读法中三位数字看成一整体，后面再加一个计数单位。从最右边往左数，三位一单位，例如12,345 等  
2.每三位数后记得带上计数单位 分别是thousand, million, billion.  
3.公式：百万以下千以上的数 X thousand X, 10亿以下百万以上的数：X million X thousand X, 10 亿以上的数：X billion X million X thousand X. 每个X分别代表三位数或两位数或一位数。  
4.在英式英语中百位数和十位数之间要加and，美式英语中则会省略，我们这个题目采用加上and，百分位为零的话，这道题目我们省略and  
  
下面再看几个数字例句：  
22: twenty two  
100: one hundred  
145: one hundred and forty five  
1,234: one thousand two hundred and thirty four  
8,088: eight thousand (and) eighty eight (注:这个and可加可不加，这个题目我们选择不加)  
486,669: four hundred and eighty six thousand six hundred and sixty nine  
1,652,510: one million six hundred and fifty two thousand five hundred and ten  
  
说明：  
数字为正整数，不考虑小数，转化结果为英文小写；  
保证输入的数据合法  
关键字提示：and，billion，million，thousand，hundred。  
  
数据范围：1 \le n \le 2000000 \1≤n≤2000000   
  
  
输入描述：  
输入一个long型整数  
  
输出描述：  
输出相应的英文写法  
  
示例1  
输入：  
22  
  
输出：  
twenty two

描述  
现有n种砝码，重量互不相等，分别为 m1,m2,m3…mn ；  
每种砝码对应的数量为 x1,x2,x3...xn 。现在要用这些砝码去称物体的重量(放在同一侧)，问能称出多少种不同的重量。  
  
  
注：  
  
称重重量包括 0  
  
数据范围：每组输入数据满足 1 \le n \le 10 \1≤n≤10 ， 1 \le m\_i \le 2000 \1≤m   
i  
  
 ≤2000 ， 1 \le x\_i \le 10 \1≤x   
i  
  
 ≤10   
输入描述：  
对于每组测试数据：  
第一行：n --- 砝码的种数(范围[1,10])  
第二行：m1 m2 m3 ... mn --- 每种砝码的重量(范围[1,2000])  
第三行：x1 x2 x3 .... xn --- 每种砝码对应的数量(范围[1,10])  
输出描述：  
利用给定的砝码可以称出的不同的重量数  
  
示例1  
输入：  
2  
1 2  
2 1  
  
输出：  
5  
  
说明：  
可以表示出0，1，2，3，4五种重量。

描述  
输入一行字符，分别统计出包含英文字母、空格、数字和其它字符的个数。  
  
数据范围：输入的字符串长度满足 1 \le n \le 1000 \1≤n≤1000   
  
输入描述：  
输入一行字符串，可以有空格  
  
输出描述：  
统计其中英文字符，空格字符，数字字符，其他字符的个数  
  
示例1  
输入：  
1qazxsw23 edcvfr45tgbn hy67uj m,ki89ol.\\/;p0-=\\][  
  
输出：  
26  
3  
10  
12

描述  
IP地址是由4个0-255之间的整数构成的，用"."符号相连。  
二进制的IP地址格式有32位，例如：10000011，01101011，00000011，00011000;每八位用十进制表示就是131.107.3.24  
子网掩码是用来判断任意两台计算机的IP地址是否属于同一子网络的根据。  
子网掩码与IP地址结构相同，是32位二进制数，由1和0组成，且1和0分别连续，其中网络号部分全为“1”和主机号部分全为“0”。  
你可以简单的认为子网掩码是一串连续的1和一串连续的0拼接而成的32位二进制数，左边部分都是1，右边部分都是0。  
利用子网掩码可以判断两台主机是否在同一子网中。  
若两台主机的IP地址分别与它们的子网掩码进行逻辑“与”运算（按位与/AND）后的结果相同，则说明这两台主机在同一子网中。  
  
示例：  
I P 地址　 192.168.0.1  
子网掩码　 255.255.255.0  
  
转化为二进制进行运算：  
  
I P 地址　 11000000.10101000.00000000.00000001  
子网掩码　11111111.11111111.11111111.00000000  
  
AND运算 11000000.10101000.00000000.00000000  
  
转化为十进制后为：  
192.168.0.0  
  
  
I P 地址　 192.168.0.254  
子网掩码　 255.255.255.0  
  
  
转化为二进制进行运算：  
  
I P 地址　11000000.10101000.00000000.11111110  
子网掩码 11111111.11111111.11111111.00000000  
  
AND运算 11000000.10101000.00000000.00000000  
  
转化为十进制后为：  
192.168.0.0  
  
通过以上对两台计算机IP地址与子网掩码的AND运算后，我们可以看到它运算结果是一样的。均为192.168.0.0，所以这二台计算机可视为是同一子网络。  
  
输入一个子网掩码以及两个ip地址，判断这两个ip地址是否是一个子网络。  
若IP地址或子网掩码格式非法则输出1，若IP1与IP2属于同一子网络输出0，若IP1与IP2不属于同一子网络输出2。  
  
注:  
有效掩码与IP的性质为：  
1. 掩码与IP每一段在 0 - 255 之间  
2. 掩码的二进制字符串前缀为网络号，都由‘1’组成；后缀为主机号，都由'0'组成  
  
输入描述：  
3行输入，第1行是输入子网掩码、第2，3行是输入两个ip地址  
题目的示例中给出了三组数据，但是在实际提交时，你的程序可以只处理一组数据（3行）。  
  
输出描述：  
若IP地址或子网掩码格式非法则输出1，若IP1与IP2属于同一子网络输出0，若IP1与IP2不属于同一子网络输出2  
  
示例1  
输入：  
255.255.255.0  
192.168.224.256  
192.168.10.4  
255.0.0.0  
193.194.202.15  
232.43.7.59  
255.255.255.0  
192.168.0.254  
192.168.0.1  
  
输出：  
1  
2  
0  
  
说明：  
对于第一个例子:  
255.255.255.0  
192.168.224.256  
192.168.10.4  
其中IP:192.168.224.256不合法，输出1  
  
对于第二个例子:  
255.0.0.0  
193.194.202.15  
232.43.7.59  
2个与运算之后，不在同一个子网，输出2  
  
对于第三个例子，2个与运算之后，如题目描述所示，在同一个子网，输出0

描述  
假设一个球从任意高度自由落下，每次落地后反跳回原高度的一半; 再落下, 求它在第5次落地时，共经历多少米?第5次反弹多高？  
  
  
数据范围：输入的小球初始高度满足 1 \le n \le 1000 \1≤n≤1000 ，且保证是一个整数  
  
输入描述：  
输入起始高度，int型  
  
输出描述：  
分别输出第5次落地时，共经过多少米以及第5次反弹多高。  
注意：你可以认为你输出保留六位或以上小数的结果可以通过此题。  
示例1  
输入：  
1  
  
输出：  
2.875  
0.03125

描述  
有一种兔子，从出生后第3个月起每个月都生一只兔子，小兔子长到第三个月后每个月又生一只兔子。  
例子：假设一只兔子第3个月出生，那么它第5个月开始会每个月生一只兔子。  
一月的时候有一只兔子，假如兔子都不死，问第n个月的兔子总数为多少？  
数据范围：输入满足 1 \le n \le 31 \1≤n≤31   
输入描述：  
输入一个int型整数表示第n个月  
  
输出描述：  
输出对应的兔子总数  
  
示例1  
输入：  
3  
  
输出：  
2

描述  
有一种技巧可以对数据进行加密，它使用一个单词作为它的密匙。下面是它的工作原理：首先，选择一个单词作为密匙，如TRAILBLAZERS。如果单词中包含有重复的字母，只保留第1个，将所得结果作为新字母表开头，并将新建立的字母表中未出现的字母按照正常字母表顺序加入新字母表。如下所示：  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
  
T R A I L B Z E S C D F G H J K M N O P Q U V W X Y (实际需建立小写字母的字母表，此字母表仅为方便演示）  
  
上面其他用字母表中剩余的字母填充完整。在对信息进行加密时，信息中的每个字母被固定于顶上那行，并用下面那行的对应字母一一取代原文的字母(字母字符的大小写状态应该保留)。因此，使用这个密匙， Attack AT DAWN (黎明时攻击)就会被加密为Tpptad TP ITVH。  
  
请实现下述接口，通过指定的密匙和明文得到密文。  
  
数据范围：1 \le n \le 100 \1≤n≤100 ，保证输入的字符串中仅包含小写字母  
  
输入描述：  
先输入key和要加密的字符串  
  
输出描述：  
返回加密后的字符串  
  
示例1  
输入：  
nihao  
ni  
  
输出：  
le

描述  
蛇形矩阵是由1开始的自然数依次排列成的一个矩阵上三角形。  
  
例如，当输入5时，应该输出的三角形为：  
  
1 3 6 10 15  
  
2 5 9 14  
  
4 8 13  
  
7 12  
  
11  
  
  
输入描述：  
输入正整数N（N不大于100）  
  
输出描述：  
输出一个N行的蛇形矩阵。  
  
示例1  
输入：  
4  
  
输出：  
1 3 6 10  
2 5 9  
4 8  
7

描述  
Lily上课时使用字母数字图片教小朋友们学习英语单词，每次都需要把这些图片按照大小（ASCII码值从小到大）排列收好。请大家给Lily帮忙，通过代码解决。  
Lily使用的图片使用字符"A"到"Z"、"a"到"z"、"0"到"9"表示。  
  
数据范围：每组输入的字符串长度满足 1 \le n \le 1000 \1≤n≤1000   
  
输入描述：  
一行，一个字符串，字符串中的每个字符表示一张Lily使用的图片。  
  
输出描述：  
Lily的所有图片按照从小到大的顺序输出  
  
示例1  
输入：  
Ihave1nose2hands10fingers  
  
输出：  
0112Iaadeeefghhinnnorsssv

描述  
原理：ip地址的每段可以看成是一个0-255的整数，把每段拆分成一个二进制形式组合起来，然后把这个二进制数转变成  
一个长整数。  
举例：一个ip地址为10.0.3.193  
每段数字 相对应的二进制数  
10 00001010  
0 00000000  
3 00000011  
193 11000001  
  
组合起来即为：00001010 00000000 00000011 11000001,转换为10进制数就是：167773121，即该IP地址转换后的数字就是它了。  
  
数据范围：保证输入的是合法的 IP 序列  
  
输入描述：  
输入   
1 输入IP地址  
2 输入10进制型的IP地址  
  
输出描述：  
输出  
1 输出转换成10进制的IP地址  
2 输出转换后的IP地址  
  
示例1  
输入：  
10.0.3.193  
167969729  
  
输出：  
167773121  
10.3.3.193

描述  
Catcher是MCA国的情报员，他工作时发现敌国会用一些对称的密码进行通信，比如像这些ABBA，ABA，A，123321，但是他们有时会在开始或结束时加入一些无关的字符以防止别国破解。比如进行下列变化 ABBA->12ABBA,ABA->ABAKK,123321->51233214　。因为截获的串太长了，而且存在多种可能的情况（abaaab可看作是aba,或baaab的加密形式），Cathcer的工作量实在是太大了，他只能向电脑高手求助，你能帮Catcher找出最长的有效密码串吗？  
  
数据范围：字符串长度满足 1 \le n \le 2500 \1≤n≤2500   
输入描述：  
输入一个字符串（字符串的长度不超过2500）  
  
输出描述：  
返回有效密码串的最大长度  
  
示例1  
输入：  
ABBA  
  
输出：  
4  
  
示例2  
输入：  
ABBBA  
  
输出：  
5  
  
示例3  
输入：  
12HHHHA  
  
输出：  
4

描述  
对字符串中的所有单词进行倒排。  
  
说明：  
  
1、构成单词的字符只有26个大写或小写英文字母；  
  
2、非构成单词的字符均视为单词间隔符；  
  
3、要求倒排后的单词间隔符以一个空格表示；如果原字符串中相邻单词间有多个间隔符时，倒排转换后也只允许出现一个空格间隔符；  
  
4、每个单词最长20个字母；  
  
数据范围：字符串长度满足 1 \le n \le 10000 \1≤n≤10000   
输入描述：  
输入一行，表示用来倒排的句子  
  
输出描述：  
输出句子的倒排结果  
  
示例1  
输入：  
I am a student  
  
输出：  
student a am I  
  
示例2  
输入：  
$bo\*y gi!r#l  
  
输出：  
l r gi y bo

描述  
按照指定规则对输入的字符串进行处理。  
  
详细描述：  
  
第一步：将输入的两个字符串str1和str2进行前后合并。如给定字符串 "dec" 和字符串 "fab" ， 合并后生成的字符串为 "decfab"  
  
第二步：对合并后的字符串进行排序，要求为：下标为奇数的字符和下标为偶数的字符分别从小到大排序。这里的下标的意思是字符在字符串中的位置。注意排序后在新串中仍需要保持原来的奇偶性。例如刚刚得到的字符串“decfab”，分别对下标为偶数的字符'd'、'c'、'a'和下标为奇数的字符'e'、'f'、'b'进行排序（生成 'a'、'c'、'd' 和 'b' 、'e' 、'f'），再依次分别放回原串中的偶数位和奇数位，新字符串变为“abcedf”  
  
第三步：对排序后的字符串中的'0'~'9'、'A'~'F'和'a'~'f'字符，需要进行转换操作。  
转换规则如下：  
对以上需要进行转换的字符所代表的十六进制用二进制表示并倒序，然后再转换成对应的十六进制大写字符（注：字符 a~f 的十六进制对应十进制的10~15，大写同理）。  
如字符 '4'，其二进制为 0100 ，则翻转后为 0010 ，也就是 2 。转换后的字符为 '2'。  
如字符 ‘7’，其二进制为 0111 ，则翻转后为 1110 ，对应的十进制是14，转换为十六进制的大写字母为 'E'。  
如字符 'C'，代表的十进制是 12 ，其二进制为 1100 ，则翻转后为 0011，也就是3。转换后的字符是 '3'。  
根据这个转换规则，由第二步生成的字符串 “abcedf” 转换后会生成字符串 "5D37BF"。  
  
  
数据范围：输入的字符串长度满足 1 \le n \le 100 \1≤n≤100   
  
输入描述：  
样例输入两个字符串，用空格隔开。  
  
输出描述：  
输出转化后的结果。  
  
示例1  
输入：  
dec fab  
  
输出：  
5D37BF  
  
示例2  
输入：  
ab CD  
  
输出：  
3B5D  
  
说明：  
合并后为abCD，按奇数位和偶数位排序后是CDab（请注意要按ascii码进行排序，所以C在a前面，D在b前面），转换后为3B5D   
示例3  
输入：  
123 15  
  
输出：  
88C4A

描述  
对输入的字符串进行加解密，并输出。  
  
加密方法为：  
  
当内容是英文字母时则用该英文字母的后一个字母替换，同时字母变换大小写,如字母a时则替换为B；字母Z时则替换为a；  
  
当内容是数字时则把该数字加1，如0替换1，1替换2，9替换0；  
  
其他字符不做变化。  
  
解密方法为加密的逆过程。  
数据范围：输入的两个字符串长度满足 1 \le n \le 1000 \1≤n≤1000 ，保证输入的字符串都是只由大小写字母或者数字组成  
输入描述：  
第一行输入一串要加密的密码  
第二行输入一串加过密的密码  
  
输出描述：  
第一行输出加密后的字符  
第二行输出解密后的字符  
  
示例1  
输入：  
abcdefg  
BCDEFGH  
  
输出：  
BCDEFGH  
abcdefg

描述  
题目描述  
若两个正整数的和为素数，则这两个正整数称之为“素数伴侣”，如2和5、6和13，它们能应用于通信加密。现在密码学会请你设计一个程序，从已有的 N （ N 为偶数）个正整数中挑选出若干对组成“素数伴侣”，挑选方案多种多样，例如有4个正整数：2，5，6，13，如果将5和6分为一组中只能得到一组“素数伴侣”，而将2和5、6和13编组将得到两组“素数伴侣”，能组成“素数伴侣”最多的方案称为“最佳方案”，当然密码学会希望你寻找出“最佳方案”。  
  
输入:  
  
有一个正偶数 n ，表示待挑选的自然数的个数。后面给出 n 个具体的数字。  
  
输出:  
  
输出一个整数 K ，表示你求得的“最佳方案”组成“素数伴侣”的对数。  
  
  
数据范围： 1 \le n \le 100 \1≤n≤100 ，输入的数据大小满足 2 \le val \le 30000 \2≤val≤30000   
输入描述：  
输入说明  
1 输入一个正偶数 n  
2 输入 n 个整数  
  
输出描述：  
求得的“最佳方案”组成“素数伴侣”的对数。  
  
示例1  
输入：  
4  
2 5 6 13  
  
输出：  
2

描述  
定义一个单词的“兄弟单词”为：交换该单词字母顺序（注：可以交换任意次），而不添加、删除、修改原有的字母就能生成的单词。  
兄弟单词要求和原来的单词不同。例如： ab 和 ba 是兄弟单词。 ab 和 ab 则不是兄弟单词。  
现在给定你 n 个单词，另外再给你一个单词 x ，让你寻找 x 的兄弟单词里，按字典序排列后的第 k 个单词是什么？  
注意：字典中可能有重复单词。  
  
数据范围：1 \le n \le 1000 \1≤n≤1000 ，输入的字符串长度满足 1 \le len(str) \le 10 \1≤len(str)≤10 ， 1 \le k < n \1≤k<n   
输入描述：  
输入只有一行。 先输入字典中单词的个数n，再输入n个单词作为字典单词。 然后输入一个单词x 最后后输入一个整数k  
输出描述：  
第一行输出查找到x的兄弟单词的个数m 第二行输出查找到的按照字典顺序排序后的第k个兄弟单词，没有符合第k个的话则不用输出。  
示例1  
输入：  
3 abc bca cab abc 1  
  
输出：  
2  
bca  
  
示例2  
输入：  
6 cab ad abcd cba abc bca abc 1  
  
输出：  
3  
bca  
  
说明：  
abc的兄弟单词有cab cba bca，所以输出3  
经字典序排列后，变为bca cab cba，所以第1个字典序兄弟单词为bca

描述  
编写一个程序，将输入字符串中的字符按如下规则排序。  
  
规则 1 ：英文字母从 A 到 Z 排列，不区分大小写。  
  
如，输入： Type 输出： epTy  
  
规则 2 ：同一个英文字母的大小写同时存在时，按照输入顺序排列。  
  
如，输入： BabA 输出： aABb  
  
规则 3 ：非英文字母的其它字符保持原来的位置。  
  
  
如，输入： By?e 输出： Be?y  
  
数据范围：输入的字符串长度满足 1 \le n \le 1000 \1≤n≤1000   
  
输入描述：  
输入字符串  
输出描述：  
输出字符串  
示例1  
输入：  
A Famous Saying: Much Ado About Nothing (2012/8).  
  
输出：  
A aaAAbc dFgghh: iimM nNn oooos Sttuuuy (2012/8).

描述  
实现删除字符串中出现次数最少的字符，若出现次数最少的字符有多个，则把出现次数最少的字符都删除。输出删除这些单词后的字符串，字符串中其它字符保持原来的顺序。  
  
数据范围：输入的字符串长度满足 1 \le n \le 20 \1≤n≤20 ，保证输入的字符串中仅出现小写字母  
输入描述：  
字符串只包含小写英文字母, 不考虑非法输入，输入的字符串长度小于等于20个字节。  
  
输出描述：  
删除字符串中出现次数最少的字符后的字符串。  
  
示例1  
输入：  
aabcddd  
  
输出：  
aaddd

描述  
某商店规定：三个空汽水瓶可以换一瓶汽水，允许向老板借空汽水瓶（但是必须要归还）。  
小张手上有n个空汽水瓶，她想知道自己最多可以喝到多少瓶汽水。  
数据范围：输入的正整数满足 1 \le n \le 100 \1≤n≤100   
  
注意：本题存在多组输入。输入的 0 表示输入结束，并不用输出结果。  
输入描述：  
输入文件最多包含 10 组测试数据，每个数据占一行，仅包含一个正整数 n（ 1<=n<=100 ），表示小张手上的空汽水瓶数。n=0 表示输入结束，你的程序不应当处理这一行。  
  
输出描述：  
对于每组测试数据，输出一行，表示最多可以喝的汽水瓶数。如果一瓶也喝不到，输出0。  
  
示例1  
输入：  
3  
10  
81  
0  
  
输出：  
1  
5  
40  
  
说明：  
样例 1 解释：用三个空瓶换一瓶汽水，剩一个空瓶无法继续交换  
样例 2 解释：用九个空瓶换三瓶汽水，剩四个空瓶再用三个空瓶换一瓶汽水，剩两个空瓶，向老板借一个空瓶再用三个空瓶换一瓶汽水喝完得一个空瓶还给老板

描述  
现在有一种密码变换算法。  
九键手机键盘上的数字与字母的对应： 1--1， abc--2, def--3, ghi--4, jkl--5, mno--6, pqrs--7, tuv--8 wxyz--9, 0--0，把密码中出现的小写字母都变成九键键盘对应的数字，如：a 变成 2，x 变成 9.  
而密码中出现的大写字母则变成小写之后往后移一位，如：X ，先变成小写，再往后移一位，变成了 y ，例外：Z 往后移是 a 。  
数字和其它的符号都不做变换。  
数据范围： 输入的字符串长度满足 1 \le n \le 100 \1≤n≤100   
输入描述：  
输入一组密码，长度不超过100个字符。  
  
输出描述：  
输出密码变换后的字符串  
  
示例1  
输入：  
YUANzhi1987  
  
输出：  
zvbo9441987