|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  . | 【问题描述】**二叉树遍历，从前序、中序到后序**  二叉树是一种非常重要的[数据结构](http://lib.csdn.net/base/datastructure)，非常多其他数据结构都是基于二叉树的基础演变而来的。对于二叉树，深度遍历有前序、中序以及后序三种遍历方法。  三种基本的遍历思想为：  前序遍历：根结点 ---> 左子树 ---> 右子树  中序遍历：左子树---> 根结点 ---> 右子树  后序遍历：左子树 ---> 右子树 ---> 根结点  比如，求以下二叉树的各种遍历  hhh.png  前序遍历：1  2  4  5  7  8  3  6  中序遍历：4  2  7  5  8  1  3  6  后序遍历：4  7  8  5  2  6  3  1  需要你编写程序解决的问题是：已知一个二叉树的前序遍历和中序遍历的结果，给出该二叉树的后序遍历的结果。  【输入形式】  有多组测试数据，每组测试数据三行，每组测试数据第一行只有一个正整数n，表示二叉树节点的数目，n=0意味着输入结束并且不需要处理。  每组测试数据第二行是二叉树的前序遍历的结果，是一个长度为n的字符串，每个节点由一个字符表示，字符是大小写英文字母及10个数字,不同的节点用不同的字符表示，也即无论前序遍历和中序遍历的字符串中没有重复的字符。  每组测试数据第二行是二叉树的中序遍历的结果，也是一个长度为n的字符串。  40%的测试数据1 ≤ n≤ 10；  30%的测试数据1 ≤ n≤ 20；  20%的测试数据1 ≤ n≤ 40；  10%的测试数据1 ≤ n≤ 62；  【输出形式】  对于每组测试数据，输出一行，是一个长度为n的字符串，表示二叉树后序遍历的结果。  【样例输入】  8  12457836  42758136  4  abcd  abcd  4  abcd  dcba  0  【样例输出】  47852631  dcba  dcba  【样例说明】 【评分标准】   |  | | --- | | **写出来吧** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **2.** | 【问题描述】  读入一个自然数n，计算其各位数字之和，用汉语拼音写出和的每一位数字。  【输入形式】  每个测试输入包含1个测试用例，即给出自然数n的值。这里保证n小于10的100次方。  【输出形式】  在一行内输出n的各位数字之和的每一位，拼音数字间有1 空格，但一行中最后一个拼音数字后没有空格。  【样例输入】  1234567890987654321123456789  【样例输出】  yi san wu  【样例说明】 友情提示汉语拼音  0~9：ling yi er san si wu liu qi ba jiu shi   |  | | --- | | **字符串数字置换** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **3.** | 【问题描述】  从键盘接收用户输入的字符串, 对用户输入的每个字符串的处理是：将字符串内的每一个十进制数字字符置换成下列表格中右边所对应的一个字符串（所有其他字符不变），然后将转换的结果显示在屏幕上；并分别计算每个数字的置换次数。   |  |  | | --- | --- | | 十进制数字字符 | 置换成 | | 0 | (Zero) | | 1 | (One) | | 2 | (Two) | | 3 | (Three) | | 4 | (Four) | | 5 | (Five) | | 6 | (Six) | | 7 | (Seven) | | 8 | (Eight) | | 9 | (Nine) |   例如，若用户输入的字符串为           Page112-Line3，  则程序5的输出是：           Page(One) (One) (Two)-Line(Three),  数字0到9的置换次数分别是  0 2 1 1 0 0 0 0 0 0  【输入形式】  输入一行字符串，其中可包含字母、数字、空格或其他符号（英文）  【输出形式】  第一行为将字符串中的数字转换为表格中的内容后输出  第二行为数字0~9被转换的次数  【样例输入】  Page112-Line3  【样例输出】  Page(One)(One)(Two)-Line(Three)  0 2 1 1 0 0 0 0 0 0 |  | |  |  |  | | **身份证校验** | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **4.** | 【问题描述】  我国国标〖GB 11643-1999〗中规定：公民身份号码是18位特征组合码，由十七位数字本体码和一位数字校验码组成。排列顺序从左至右依次为：六位数字地址码，八位数字出生日期码，三位数字顺序码和一位数字校验码。其校验码(最后一位)计算方法和步骤为：  (1)十七位数字本体码加权求和公式  S = Sum(Ai \* Wi), i = 0, ... , 16 ，先对前17位数字的权求和  其中Ai：表示第i位置上的身份证号码数字值  Wi：表示第i位置上的加权因子，前17位加权因子从左到右分别为  Wi：7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2  (2)计算模  Y = mod(S, 11)  (3)通过模Y查下表得到对应的校验码   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 校验码 | 1 | 0 | X | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |   例如：某身份证前17位为11010519491231002   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | | wi | 7 | 9 | 10 | 5 | 8 | 4 | 2 | 1 | 6 | 3 | 7 | 9 | 10 | 5 | 8 | 4 | 2 | |  | 1 | 1 | 0 | 1 | 0 | 5 | 1 | 9 | 4 | 9 | 1 | 2 | 3 | 1 | 0 | 0 | 2 | | 积 | 7 | 9 | 0 | 5 | 0 | 20 | 2 | 9 | 24 | 27 | 7 | 18 | 30 | 5 | 0 | 0 | 4 |   得到和为：167；则模为y=167%11=2  查(3)得校验码为X(大写）  请按上面所述步骤编程，输入一个二代身份证号，检查该身份证是否正确。  【输入形式】  输入若干行，每行一个身份证号码，最后一行输入-1  【输出形式】  输出1代表正确，0代表错误  【样例输入】  120223198902021249  130132199210293822  130402198207290622  -1  【样例输出】  1  1  0 | | **多项式加法** | |  |  |  |  | | --- | --- | --- | | **5.** | 【问题描述】  一个多项式可以表示为一组数对，数对中第一个数始终为整数，且唯一，表示多项式的次数，另一数表示为对应的系数且不为0。输入两组数对，每组以0 0作为结束，实现对两个多项式的加法并按降幂输出结果数对  【输入形式】  每行输入一个数对，以空格为分隔符，以0 0结束  【输出形式】  每行输出一个数对，以空格为分隔符  【样例输入】  5 12  3 8  1 2  15 5  0 10  0 0  3 12  30 1  15 5  0 0  【样例输出】  30 1  15 10  5 12  3 20  1 2  0 10 |  | |  |  |  | | -**挖掘机技术哪家强** | | |  |  |  | | --- | --- | | **6.** | 【问题描述】  为了用事实说明挖掘机技术到底哪家强，组织一场挖掘机技能大赛。现请你根据比赛结果统计出技术最强的那个学校。  【输入形式】  输入在第1行给出不超过105的正整数N，即参赛人数。随后N行，每行给出一位参赛者的信息和成绩，包括其所代表的学校的编号、及其比赛成绩（百分制），中间以空格分隔。  【输出形式】  在一行中给出总得分最高的学校的编号、及其总分，中间以空格分隔。题目保证答案唯一，没有并列。  【样例输入】  6 3 65 2 80 1 100 2 70 3 40 3 0  【样例输出】  2 150  【问题说明】  建议练习使用STL中的map | |  | |  | |  | | --- | | **数码管** |  |  |  | | --- | --- | | **7.** | 【问题描述】  液晶数码管用七笔阿拉数字表示的十个数字，把横和竖的一 个短划都称为一笔，即７有３笔，８有７笔等。对于十个数字一种排列，要做到两相邻数字都可以由另一个数字加上几笔或减去几笔组成，但不能又加又减。比如 ７→３是允许的，７→２不允许。任意输入一组数，判断是否符合上述规则，注意，1在右边。  【输入形式】  每行输入一个0~9的排列，数字之间用空格分隔，以-1作为输入结束  【输出形式】  输出YES或NO  【样例输入】  4 1 0 7 3 9 5 6 8 2  3 5 1 6 2 7 9 0 4 8  -1  【样例输出】  YES  NO | | **数塔** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **8.** | 【问题描述】  给定一个数塔，如下图所示。在此数塔中，从顶部出发，在每一节点可以选择走左下或右下，一直走到底层。请找出一条路径，使路径上的数值和最大。   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  | 9 |  |  |  |  | |  |  |  | 12 |  | 15 |  |  |  | |  |  | 10 |  | 6 |  | 8 |  |  | |  | 2 |  | 18 |  | 9 |  | 5 |  | | 19 |  | 7 |  | 10 |  | 4 |  | 16 |   【输入形式】  输入时第一行一个整数n，表示该数塔的行数，其余n行表示该塔每行的数值  【输出形式】  输出包含两行，第一行为最大路径上的数值之和， 第二行n个数字为从上而下最大路径数值  【样例输入】  5  9  12 15  10 6 8  2 18 9 5  19 7 10 4 16  【样例输出】  59  9 12 10 18 10 | | **日历问题** | |  |  |  | | --- | --- | | **9.** | 【问题描述】  在我们现在使用的日历中, 闰年被定义为能被4整除的年份，但是能被100整除而不能被400整除的年是例外，它们不是闰年。例如：1700, 1800, 1900 和 2100 不是闰年，而 1600, 2000 和 2400是闰年。 给定从公元2000年1月1日开始逝去的天数，你的任务是给出这一天是哪年哪月哪日星期几。  【输入形式】  输入包含若干行，每行包含一个正整数，表示从2000年1月1日开始逝去的天数。输入最后一行是−1, 不必处理。可以假设结果的年份不会超过9999。  【输出形式】  对每个测试样例，输出一行，该行包含对应的日期和星期几。格式为“YYYY-MM-DD DayOfWeek”, 其中 “DayOfWeek” 必须是下面中的一个： "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday" and "Saturday“。  【样例输入】  1730  1740  1750  1751  -1  【样例输出】  2004-09-26 Sunday  2004-10-06 Wednesday  2004-10-16 Saturday  2004-10-17 Sunday | | **斯诺克台球** | |  |  |  | | --- | --- | | **10.** | 【问题描述】  斯诺克台球是一项古老而又时尚的运动，使用长方形球桌，台面四角以及两长边中心位置各有一个球袋，使用的球分为1个白球，15个红球和6个彩球共22个球。  其中母球（白球）1只，目标球21只。目标球中：红球15只各1分、黄球1只2分、绿球1只3分、咖啡球1只4分、蓝球1只5分、粉球1只6分、黑球1只7分。  djh.jpg  选手需要使用球杆撞击母球去击打目标球来完成得分，每局开始时总是先从红球开始。击球顺序为先打进红球（每次击打允许多个红球同时落袋），然后任意指定一个目标彩球击打，如果该彩球被打进（打进后需要再摆回），然后接着击打红球，直到红球全部落袋，然后以黄、绿、咖啡、蓝、粉红、黑的顺序逐个击球（不再摆回），最后以得分高者为胜。任何时候红球落袋都不再摆回，任何时候因犯规导致彩球落袋，彩球必须摆回。  斯诺克比赛由双方轮流击打，必须击打合规的目标球，打进则本方得到相应的分数并继续击打，未打进或犯规轮换为对方击打，未打进不得分，犯规将进行罚分处理。  犯规规则如下：  1.     当击打目标球时，如果先击打到或同时击打到一个或多个其他颜色的球，或者有其他颜色的球落袋，或者打空(未击打到任何球)，则视为犯规。此时需要比较目标球的分值和与本犯规相关的其他颜色的球的分值，取其中最高的分值，如果该分值小于4，则对方加4分，否则对方加该分值。  2.     当击打红球落袋后，继续击打任意彩球时打空，即未打击到任何球，对方加4分。  相比正式的斯诺克比赛，本问题对规则进行了简化，任何时候都可以结束比赛并计算比赛结果，不考虑白球落袋的情况。  信息化时代的智能台球桌能自动记录实际比赛时的击打记录，并传送到后台，但该记录仅仅是流水记录，并且无参赛选手的任何信息，需要你编程计算每场比赛的比分，同时需要计算单杆100分及以上的情况（单杆得分是指选手一次连续击打所得分数之和）。  【输入形式】  输入第一行为正整数t (t≤100)，表示有t组测试数据，每组数据代表一局比赛。  在输入中，球的颜色表示为：  **r-红色球 y-黄色球 g-绿色球 c-咖啡色球 b-蓝色球 p-粉红球 B-黑色球**  接下来的每组数据包括若干行，每一行为一次击打的结果，为智能球桌记录下来的流水记录，每组数据最后一行为-1，表示每组数据的结束。  流水记录包含用空格分隔的2个部分：  首先撞到的球 落袋球及数量  第一部分“首先撞到的球”为一个字符串，可以是“rygcbpB”中1个或多个字符组合（可能有多个字符“r”）,或为字符串“NULL”。为“NULL”时，第二部分必为空，表示该次击打未撞击到任何球也没有任何球落袋。当红球落袋后继续击打任意彩球时，该部分为“ygcbpB”中的任意单个字符时都认为是合规的目标球。  第二部分“落袋球及数量”为一个字符串，例如“r2gb”，代表本次击打有两个红球落袋，以及绿球和篮球落袋，红色球r后面有数字（大于0小于16），表示红球的落袋数，其他彩球后无数字。该部分可以为空，表示本次击打无球落袋。  比赛在A与B之间进行，每局比赛总是由A先开球。  【输出形式】  输出为t+1行，前t行每行输出用冒号分隔的两个整数，表示每局比赛A与B之间的比分；最后一行输出用冒号分隔的两个整数，表示t局比赛之后A与B之间获得的单杆100分及以上的次数之比（单杆得分是指选手一次连续击打所得分数之和）。  【样例输入】  3  r r1  B  r r2  c c  r r1  b g  -1  rp r1  r br2B  NULL  r r12  y y  g p  -1  rr r3  NULL  r r1  yg y  -1  【样例输出】  6:7  13:24  7:5  0:0  【样例说明】  第一局比赛：  A击打红球，打进1个红球，得1分，比分为 1:0  A继续击打任意彩球，打到黑球，未打进，不得分，比分为1:0  轮换为B击打红球，打进两个红球，得2分，比分为1:2  B继续击打任意彩球，打到咖啡球，打进咖啡球，咖啡球摆回，得4分，比分为1:6  B继续击打红球，打进一个红球，得1分，比分为1:7  B继续击打任意彩球，打到蓝球，打进绿球，犯规，取分值最大者蓝球，绿球摆回，对方加5分，比分为6:7  -1比赛结束  第二局比赛：  A击打红球，首先打到红球和粉球，犯规，打进1个红球和咖啡球，犯规，咖啡球摆回，取分值最大的粉球，对方加6分，比分为0:6  B击打红球，首先打到红球，打进蓝球、2个红球和黑球，犯规，蓝球和黑球摆回，取分值最大的黑球，对方加7分，比分为7:6  A击打红球，未打到任何球，犯规，对方加4分，比分为7:10  B击打红球，打到红球，打进12个红球，加12分，比分为7:22  B击打任意彩球，打到黄球，打进黄球，黄球摆回，得2分，比分为7:24  B击打黄球，打到绿球，打进粉球，犯规，粉球摆回，对方加6分，比分为13:24  -1比赛结束  第三局比赛：  A击打红球，打到2个红球，打进3个红球，加3分，比分为3:0  A击打任意彩球，打空，未打到任何球，对方加4分，比分为3:4  B击打红球，打到1个红球，打进1个红球，加1分，比分为3:5  B击打任意彩球，打到黄球和绿球，打进黄球，犯规，黄球摆回，取分值最高的绿球，绿球分值小于4，对方加4分，比分为7:5  -1比赛结束  3局比赛中无人单杆得分过100，最后一行输出0:0  【评分标准】 | | **恺撒Caesar密码** | |  |  |  |  | | --- | --- | --- | | **11.** | 【问题描述】  Julius Caesar 生活在充满危险和阴谋的年代。为了生存，他首次发明了密码，用于军队的消息传递。假设你是Caesar 军团中的一名军官，需要把Caesar 发送的消息破译出来、并提供给你的将军。消息加密的办法是：对消息原文中的每个字母，分别用该字母之后的第5个字母替换（例如：消息原文中的每个字母A都分别替换成字母F），其他字符不 变，并且消息原文的所有字母都是大写的。 密码字母：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 原文字母：V W X Y Z A B C D E F G H I J K L M N O P Q R S T U  【输入形式】  最多不超过100个数据集组成。每个数据集由3部分组成：起始行：START 密码消息：由1到200个字符组成一行，表示Caesar发出的一条消息结束行：END 在最后一个数据集之后，是另一行：ENDOFINPUT  【输出形式】  每个数据集对应一行，是Caesar 的原始消息。  【样例输入】  START  NS BFW, JAJSYX TK NRUTWYFSHJ FWJ YMJ WJXZQY TK YWNANFQ HFZXJX  END  START  N BTZQI WFYMJW GJ KNWXY NS F QNYYQJ NGJWNFS ANQQFLJ YMFS XJHTSI NS WTRJ  END  START  IFSLJW PSTBX KZQQ BJQQ YMFY HFJXFW NX RTWJ IFSLJWTZX YMFS MJ  END  ENDOFINPUT  【样例输出】  IN WAR, EVENTS OF IMPORTANCE ARE THE RESULT OF TRIVIAL CAUSES  I WOULD RATHER BE FIRST IN A LITTLE IBERIAN VILLAGE THAN SECOND IN ROME  DANGER KNOWS FULL WELL THAT CAESAR IS MORE DANGEROUS THAN HE |  | |  |  |  | | **世界杯来了** | | |  |  |  | | --- | --- | | **12.** | 【问题描述】         2018年俄罗斯世界杯结束了，法国获得冠军，全世界球迷度过了一个非常愉快的夏天。作为中国球迷，不能总是看别人踢球，这不福利来了，根据FIFA（国际足联）及全体成员协会的一致决定，2118年世界杯将在中国举办，作为东道主，中国队将无需参加预选赛而直接参加决赛阶段的比赛。     比赛规则如下：   * 总共n（n为偶数）个球队参加比赛 * 按照分组赛积分排名，前n/2的球队进入淘汰赛 * 积分排名的规则如下：球队获胜得3分，平局得1分，失利得0分，按照积分递减、净胜球递减以及进球数递减方式排名   编写一个程序，根据给出的参赛队伍名单和所有比赛的结果，找出成功进入淘汰赛阶段的球队名单。  【输入形式】         第一行输入包含唯一整数n(1<=n<=50)，参加世界杯决赛的球队数量。接下来的n行是各球队的名字，为长度不超过30个字符的英文字符。接下来的n\*(n-1)/2行，每行格式name1-name2 num1:num2（0<=num1, num2<=100），表示对阵球队及比分.  【输出形式】         输入n/2行，表示进入淘汰赛阶段的球队，按照字典序进行排列，每个球队名字占一行。 【样例输入】  4 A B C D A-B 1:1 A-C 2:2 A-D 1:0 B-C 1:0 B-D 0:3 C-D 0:3  【样例输出】  A D  【样例说明】 【评分标准】 | |  | |
| **组个最小数** | |

|  |  |
| --- | --- |
| **13.** | 【问题描述】  给定数字0-9各若干个。你可以以任意顺序排列这些数字，但必须全部使用。目标是使得最后得到的数尽可能小（注意0不能做首位）。例如：给定两个0，两个1，三个5，一个8，我们得到的最小的数就是10015558。  现给定数字，请编写程序输出能够组成的最小的数。  【输入形式】  每个输入包含1个测试用例。每个测试用例在一行中给出多个（不超过50个）数字（0~9之间），整数间用一个空格分隔，且至少拥有1个非0的数字。  【输出形式】  在一行中输出能够组成的最小的数。  【样例输入】  2 2 0 0 0 3 0 0 1 0  【样例输出】  1000000223  【样例说明】 【评分标准】 |
| -**F1方程式冠军** | |

|  |  |
| --- | --- |
| **14.** | 【问题描述】    一级方程式F1锦标赛由一系列称为大奖赛的分站赛组成。每一场比赛的车手都根据他们的最后位置获得积分。只有前10名车手按以下顺序获得分数：25、18、15、12、10、8、6、4、2、1。在锦标赛结束时，得分最多的车手是冠军。如果有平分，则冠军是赢的最多的人（即排位第一）。如果还是平分，则选择得到排位第二最多的人，依此类推，直到没有更多的排位进行比较。    去年又提出了另一个得分制度，但被否决了。其中冠军是赢得最多的。如果有平手，冠军是得分最多的。如果仍然存在平手，则按原来的得分制度进行，即比较第二、第三、第四、...排位的次数。    在本赛季，你会得到所有比赛的结果，你将根据两个得分系统来分别确定冠军。数据保证两套系统都能得到唯一的冠军。  【输入形式】    第一行一个整数t（1<=t<=20），t是分站赛的场次数。之后是每个分站赛的最终排位情况，每个的第一行一个整数n(1<=n<=100)表示排位车手人数，之后n行按排位列出车手的名字，排位从第一到最后，车手的名字为长度不超过50的英文字符，大小写区分。 【输出形式】、    输出为两行，第一行为按照原始规则确定的冠军，第二行是按照可选规则确定的冠军。  【样例输入】  3 3 apple banana pear 2 pear banana 2 apple banana  【样例输出】  banana apple  【样例说明】 【评分标准】 |
| **选美比赛** | |

|  |  |
| --- | --- |
| **15.** | 【问题描述】  在选美大奖赛的半决赛现场，有n名选手（2<n<100）参加比赛。比赛结束时，要在现场按照选手的出场顺序宣布最后名次，获得相同分数的选手具有相同的名次，名次连续编号，不用考虑同名次的选手人数。如：  选手数量：  7  选手得分：  5，3，4，7，3，5，6  宣布名次：  3，5，4，1，5，3，2  请编程帮助大奖赛组委会完成半决赛的评分排名工作。  【输入形式】  选手数量：7  选手得分：5 3 4 7 3 5 6  【输出形式】  选手的排名：3 5 4 1 5 3 2  【样例输入】  7  5 3 4 7 3 5 6  【样例输出】  3 5 4 1 5 3 2  【样例说明】  本题的关键在于如何处理同分数的选手排名问题 |
| **到底买不买** | |

|  |  |  |
| --- | --- | --- |
| **16.** | 小红想买些珠子做一串自己喜欢的珠串。卖珠子的摊主有很多串五颜六色的珠串，但是不肯把任何一串拆散了卖。于是小红要你帮忙判断一下，某串珠子里是否包含了全部自己想要的珠子？如果是，那么告诉她有多少多余的珠子；如果不是，那么告诉她缺了多少珠子。  为方便起见，我们用[0-9]、[a-z]、[A-Z]范围内的字符来表示颜色。例如在图1中，第3串是小红想做的珠串；那么第1串可以买，因为包含了全部她想要的珠子，还多了8颗不需要的珠子；第2串不能买，因为没有黑色珠子，并且少了一颗红色的珠子。  pimg1030_1.jpg  【输入形式】  每个输入包含1个测试用例。每个测试用例分别在2行中先后给出摊主的珠串和小红想做的珠串，两串都不超过1000个珠子。  【输出形式】  如果可以买，则在一行中输出“Yes”以及有多少多余的珠子；如果不可以买，则在一行中输出“No”以及缺了多少珠子。其间以1个空格分隔。  【样例输入】  ppRYYGrrYBR2258 YrR8RrY  【样例输出】  Yes 8  【样例说明】 【评分标准】 |  |
|  |  |  |
| **蛇行矩阵** | | |

|  |  |
| --- | --- |
| **17.** | 【问题描述】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  【输入形式】 正整数 N表示层数，N 不大于 100  【输出形式】输出一个 N 行的蛇形矩阵，矩阵三角中同一行的数字用一个空格分开，行尾不要多余的空格。  【样例输入】  5  【样例输出】  1 3 6 10 15 2 5 9 14 4 8 13 7 12 11  【样例说明】 【评分标准】 |
| -**IP地址** | |

|  |  |
| --- | --- |
| **18.** | 【问题描述】  一个IP地址由32位二进制的数组成，比如：  111111111111111111111111000000002  为了便于记忆，我们将8个二进制位用一个十进制数表示，一个IP地址由四个十进制数表示，上述的IP地址表示为：  255.255.255.0  现在给你一个上述形式的IP地址，请回答IP地址的32个二进制位中，有多少位是1。  如IP地址为255.255.255.0，其中24位是1。  【输入形式】  有多组测试数据。  测试数据第一行是一个正整数T，表示测试数据组数。  每组测试数据是一个IP地址，形式为：  IP1.IP2.IP3.IP4  其中0 ≤IP1,IP2,IP3,IP4≤ 255,用十进制表示。每个IP地址不保证是实用IP地址。  40%的测试数据组数T  10≤T≤ 102；  30%的测试数据组数T  102≤T≤ 103；  20%的测试数据组数T  103≤T≤ 104；  10%的测试数据组数T  104≤T≤ 105；  【输出形式】  对于每个IP地址，输出一行包含一个非负整数：该IP地址的32个二进制位中，1的位数。  【样例输入】  5  255.255.255.0  127.0.0.1  0.0.0.1  1.2.3.4  0.0.0.0  【样例输出】  24  8  1  5  0  提示：样例中32位的IP地址为：  111111111111111111111111000000002  011111110000000000000000000000012  000000000000000000000000000000012  000000010000001000000011000001002  000000000000000000000000000000002 |
| **约瑟夫问题** | |

|  |  |
| --- | --- |
| **19.** | 【问题描述】  约瑟夫和他的小伙伴--共N个人--围成一个圈子。按照圈子的位置，每个人依次都有一个编号：从1到N。最后一个人N和第一个人1号相连。  现在他们要选出一个首领来。选择方法是：从1号开始，每隔M个人淘汰一个人，最后剩下的人就是首领。  比如有7个人，商定M为3，即每隔3人淘汰1人。那么，首先淘汰3号，然后依次淘汰6号，2号，7号，5号，1号，最后剩下的4号就是首领。  约瑟夫希望做这个首领。那么，在给定N与M后，你能告诉约瑟夫他必须是多少号才能做首领吗？  【输入形式】  输入第一行正整数t (10 ≤ t ≤ 100)，表示有多少组测试数据。  后面有t行，每行两个正整数N，M（2≤M≤20），中间用一个空格隔开。  40%的测试数据1 ≤ N≤ 10；  30%的测试数据1 ≤ N≤ 102；  20%的测试数据1 ≤ N≤ 103；  10%的测试数据1 ≤ N≤ 104；  【输出形式】  对于每组测试数据，输出一个正整数占一行：首领的号码。  【样例输入】  4  1 3  7 3  2 3  4 2  【样例输出】  1  4  2  1  【样例说明】 【评分标准】 |
| **错误的里程表** | |

|  |  |
| --- | --- |
| **20.** | 【问题描述】  三月八日，小明买了台新车。但很快小明发现汽车的里程表有问题：里程表上每一位都不显示数字3和数字8，也就是说直接从数字2跳到数字4，直接从数字7跳到数字9。小明纳闷：这车到底行驶里程是多少。  现在，小明向你求助：根据里程表显示的数字，给出真实的行驶里程。  【输入形式】  输入有多组测试数据。  输入第一行正整数T，表示有多少组测试数据。  后面有T行，每行一个非负整数，表示里程表显示数字，里面不含有数字3和8。该数字不超过10位。  40%的测试数据组数T  10≤T≤ 102；  30%的测试数据组数T  102≤T≤ 103；  20%的测试数据组数T  103≤T≤ 104；  10%的测试数据组数T  104≤T≤ 105；  【输出形式】  对于每组测试数据，输出一个整数占一行：真实的行程里程。  【样例输入】  6  0  1  12  159  111224459  124567976  【样例输出】  0  1  10  103  19212007  21913077  【样例说明】 【评分标准】 |
| **合法的括号串** | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **21.** | 【问题描述】  一个合法的括号串，是指只包含括号的串，如果满足如下条件：  （1）<> () [] {} 这四对括号是合法的；  （2）如果r是合法括号串，则<r> (r) [r] {r}也是；  （3）如果r，s是合法括号串，则rs也是；  所以<<>> , [<>{}(())],[({<>})]是合法的括号串，而)(,[( ])就不是。  【输入形式】  输入第一行正整数t (10 ≤ n ≤ 100)，表示有多少组测试数据。  后面有t行，每行一个只包含8种括号符号的括号串。  40%的括号串的长度L 2 ≤ L≤ 20；  30%的括号串的长度L 2 ≤ L≤ 200；  20%的括号串的长度L 2 ≤ L≤ 2000；  10%的括号串的长度L 2 ≤ L≤ 20000；  【输出形式】  对于每组测试数据，如果括号串是合法的，输出“**Yes**”（输出没有引号）占一行，否则，输出“**No**”（输出没有引号）占一行。  【样例输入】  6  <<>>  )(  [<>{}(())]  [({<>})]  [(])  <([{  【样例输出】  Yes  No  Yes  Yes  No  No  【样例说明】   |  | | --- | | **计算校验码** |  |  |  |  | | --- | --- | --- | | **22.** | 【问题描述】  传送一个B（B≤16）进制的数值N时，最后加上一个一位（B进制的）校验码，使得N加上校验位后能被B-1整除。比如十进制的数值12310，其校验码就是3，因为十进制数值123310能被9整除。16进制的数7816，其校验码为0，因为16进制的78016是15的倍数。超过十进制后，用字母a表示10，字母b表示11，字母c表示12，字母d表示13，字母e表示14，字母f表示15。  告诉你进制B，以及一个B进制的正整数N，要求你计算正整数N在B进制下的校验码。  【输入形式】  输入第一行正整数t (10 ≤ n ≤ 100)，表示有多少组测试数据。  后面有t行，每行两个正整数B，N（2≤ B≤16），中间用一个空格隔开，B是10进制整数，N用B进制形式表示。测试数据保证没有非法的B进制数N（也即N中每一位都是在0到B-1之间，没有前导0）。  40%的测试数据N的位数L 1 ≤ L≤ 10；  30%的测试数据N的位数L 1 ≤ L≤ 102；  20%的测试数据N的位数L 1 ≤ L≤ 103；  10%的测试数据N的位数L 1 ≤ L≤ 104；  【输出形式】  对于每组测试数据，输出一位占一行：正整数N在B进制下的校验码。（如果校验码可以为B-1，也可以为0，输出0）。  【样例输入】  4  10 123  16 78  16 1234321  12 ab  【样例输出】  3  0  e  1  【样例说明】  第一行的4表示有4组测试数据，下面四行，每行一组测试数据。  第一组测试数据 10进制数123 最后添加检验码3，10进制数1233是9（=10-1）的倍数  第二组测试数据 16进制数78 最后添加检验码0，16进制数780是15（=16-1）的倍数  第三组测试数据 16进制数1234321 最后添加检验码e（=14），16进制数1234321e是15（=16-1）的倍数  第四组测试数据 12进制数ab 最后添加检验码1，12进制数ab1是11（12-1）的倍数  【Tips】  B进制的数能被B-1整除，当且仅当各位数字和能被B-1整除。  第一组测试数据 10进制数123 最后添加检验码3，10进制数1233各位数字和是9，是9的倍数  第二组测试数据 16进制数78 最后添加检验码0，16进制数780各位数字和是15，是15的倍数  第三组测试数据 16进制数1234321 最后添加检验码e，16进制数1234321e各位数字和是30，是15的倍数  第四组测试数据 12进制数ab 最后添加检验码1，12进制数ab1各位数字和是22，是11的倍数 |  | |  |  |  | |
| **拳王阿里** | |

|  |  |  |
| --- | --- | --- |
| **23.** | 【问题描述】         阿里是上个世纪美国最著名的拳击手，阿里在20年的时间里多次获得重量级拳王称号。不过不幸的是，他在之后患上了帕金森氏病。他参加了许多比赛，多到连自己都数不清了。     有这么一段时间，他总是参加各种不同的比赛，以至于他自己也不知道从第一场比赛开始到最后一场比赛结束到底用了多长时间。他只记得比赛的第一天是星期几（S)，最后一场比赛的最后一天是星期几(E)。他还记得比赛的总天数（包括第一和最后一天）不少于L天，也不多于R天。给定S和E，能否唯一确定参加比赛总天数(包含该段时间内比赛间的间隔天数)？  【输入形式】         输入的第一行包含一个整数T，代表测试数据的组数。接下来是 T 组数据。每组数据仅有一行，首先包含两个字符串S和E，然后包含两个整数L和R。      • 1 ≤ T ≤ 10,000        1 ≤ L ≤ R ≤ 100      • S, E ∈ {“monday”,“tuesday”,“wednesday”,“thursday”,“friday”,“saturday”, “sunday”}  【输出形式】      对于每组数据：   * + 如果不存在满足条件的天数，输出一行“impossible”；   + 如果存在多个满足条件的天数，输出一行“many”；   + 否则，输出一行，包含一个整数，代表唯一满足条件的天数。   【样例输入】  3  saturday sunday 2 4  monday wednesday 1 20  saturday sunday 3 5  【样例输出】  2  many  impossible  【样例说明】  【评分标准】 |  |
|  |  |  |
| **相同生日** | | |

|  |  |
| --- | --- |
| **24.** | 【问题描述】  在一个有200人的大班级中，存在两个人生日相同的概率非常大，现给出每个学生的学号，出生月日，试找出所有生日相同的学生。  【输入形式】  第一行为整数n，表示有n个学生，n<=200。此后每行包含一个字符串和两个整数，分别表示学生的学号(字符串长度为11位)和出生月(1<=m<=12)日(1<=d<=31)，学号、月、日之间用一个空格分隔。  【输出形式】  对每组生日相同的学生，输出一行，其中前两个数字表示月和日，后面跟着所有在当天出生的学生的学号，数字、学号之间都用一个空格分隔。对所有的输出，要求按日期从前到后的顺序输出。对生日相同的学号，按输入的顺序输出。  【样例输入】  6  07101020105 3 15  07101020115 4 5  07101020118 3 15  07101020108 4 5  07101020111 4 5  07101020121 8 10  【样例输出】  3 15 07101020105 07101020118  4 5 07101020115 07101020108 07101020111  8 10 07101020121 |
| -**个位数统计** | |

|  |  |
| --- | --- |
| **25.** | 【问题描述】  给定一个k位整数N = dk-1\*10k-1 + ... + d1\*101 + d0 (0<=di<=9, i=0,...,k-1, dk-1>0)，请编写程序统计每种不同的个位数字出现的次数。例如：给定N = 100311，则有2个0，3个1，和1个3。  【输入形式】  每个输入包含1个测试用例，即一个不超过1000位的正整数N。  【输出形式】  对N中每一种不同的个位数字，以D:M的格式在一行中输出该位数字D及其在N中出现的次数M。要求按D的升序输出  【样例输入】  100311  【样例输出】  0:2  1:3  3:1 |
| **最简单的计算机** | |

|  |  |
| --- | --- |
| **26.** | 【问题描述】  一个名叫是 PigHeadThree 的研究组织设计了一台实验用的计算机，命名为 PpMm。PpMm只能执行简单的六种命令 A，B，C，D，E，F；只有二个内存 M1，M2；三个寄存器 R1，R2，R3。六种命令的含义如下： 命令 A：将内存 M1 的数据装到寄存器 R1 中； 命令 B：将内存 M2 的数据装到寄存器 R2 中； 命令 C：将寄存器 R3 的数据装到内存 M1 中； 命令 D：将寄存器 R3 的数据装到内存 M2 中； 命令 E：将寄存器 R1 中的数据和寄存器 R2 中的数据相加，结果放到寄存器 R3 中； 命令 F：将寄存器 R1 中的数据和寄存器 R2 中的数据相减，结果放到寄存器 R3 中。 你的任务是：设计一个程序模拟 PpMm 的运行。  【输入形式】  有若干组，每组有 2 行，第一行是 2 个整数，分别表示 M1 和 M2 中的初始内容；第二行是一串长度不超过 200 的由大写字母 A 到 F 组成的命令串，命令串的含义如上所述。  【输出形式】  对应每一组的输入，输出只有一行，二个整数，分别表示 M1，M2 的内容；其中 M1 和 M2 之间用逗号隔开。  【样例输入】  100 288  ABECED  876356 321456  ABECAEDBECAF  【样例输出】  388,388  2717080,1519268 |
| **小希的数表** | |

|  |  |
| --- | --- |
| **27.** | 【问题描述】  Gardon 昨天给小希布置了一道作业，即根据一张由不超过 5000 的 N(3<=N<=100)个正整数组成的数表两两相加得到 N\*(N-1)/2 个和，然后再将它们排序。例如，如果数表里含有四个数 1，3，4，9，那么正确答案是 4，5，7，10，12，13。小希做完作业以后出去玩了一阵，可是下午回家时发现原来的那张数表不见了，好在她做出的答案还在，你能帮助她根据她的答案计算出原来的数表么？  【输入形式】  包含多组数据，每组数据以一个 N 开头，接下来的一行有按照大小顺序排列的 N\*(N-1)/2 个数，是小希完成的答案。文件最后以一个 0 结束。 假设输入保证解的存在性和唯一性。  【输出形式】  对于每组数据，输出原来的数表。它们也应当是按照顺序排列的。  【样例输入】  4  4 5 7 10 12 13  4  5 6 7 8 9 10  0  【样例输出】  1 3 4 9  2 3 4 6 |
| **锤子剪刀布** | |

|  |  |
| --- | --- |
| **28.** | 【问题描述】  大家应该都会玩“锤子剪刀布”的游戏。现给出两人的交锋记录，请统计双方的胜、平、负次数，并且给出双方分别出什么手势的胜算最大。  【输入形式】  输入第1行给出正整数N（<=105），即双方交锋的次数。随后N行，每行给出一次交锋的信息，即甲、乙双方同时给出的的手势。C代表“锤子”、J代表“剪刀”、B代表“布”，第1个字母代表甲方，第2个代表乙方，中间有1个空格。  【输出形式】  输出第1、2行分别给出甲、乙的胜、平、负次数，数字间以1个空格分隔。第3行给出两个字母，分别代表甲、乙获胜次数最多的手势，中间有1个空格。如果解不唯一，则输出按字母序最小的解。  【样例输入】  10  C J  J B  C B  B B  B C  C C  C B  J B  B C  J J  【样例输出】  5 3 2  2 3 5  B B |
| **成绩大排队** | |

|  |  |
| --- | --- |
| **29.** | 【问题描述】  读入n名学生的姓名、学号、成绩，分别输出成绩最高和成绩最低学生的姓名和学号。  【输入形式】  每个测试输入包含1个测试用例，格式为   第1行：正整数n    第2行：第1个学生的姓名 学号 成绩    第3行：第2个学生的姓名 学号 成绩    ... ... ...    第n+1行：第n个学生的姓名 学号 成绩  其中姓名和学号均为不超过20个字符的字符串，成绩为0到100之间的一个整数，这里保证在一组测试用例中没有两个学生的成绩是相同的。  【输出形式】  对每个测试用例输出2行，第1行是成绩最高学生的姓名和学号，第2行是成绩最低学生的姓名和学号，字符串间有1空格。  【样例输入】  3  Joe Math990112 89  Mike CS991301 100  Mary EE990830 95  【样例输出】  Mike CS991301  Joe Math990112 |
| **最少钱币数** | |

|  |  |
| --- | --- |
| **30.** | 【问题描述】这是一个古老而又经典的问题。用给定的几种钱币凑成某个钱数，一般而言有多种方式。例如：给定了 6 种钱币面值为 2、5、10、20、50、100，用来凑 15 元，可以用 5 个 2 元、1个 5 元，或者 3 个 5 元，或者 1 个 5 元、1个 10 元，等等。显然，最少需要 2 个钱币才能凑成 15 元。         你的任务就是，给定若干个互不相同的钱币面值，编程计算，最少需要多少个钱币才能凑成某个给出的钱数。  【输入形式】输入可以有多个测试用例。每个测试用例的第一行是待凑的钱数值 M（1 <= M<= 2000，整数），接着的一行中，第一个整数 K（1 <= K <= 10）表示币种个数，随后是 K个互不相同的钱币面值 Ki(1 <= Ki <= 1000)。输入 M=0 时结束。  【输出形式】每个测试用例输出一行，即凑成钱数值 M 最少需要的钱币个数。如果凑钱失败，输出“Impossible”。你可以假设，每种待凑钱币的数量是无限多的。  【样例输入】  15 6 2 5 10 20 50 100 1 1 2 0  【样例输出】  2 Impossible  【样例说明】 【评分标准】 |
| **电能消耗** | |

|  |  |
| --- | --- |
| **31.** | 【问题描述】        汤姆对他最喜欢的笔记本电脑的耗电量很感兴趣。他的笔记本电脑有三种模式。在正常模式下，笔记本电脑每分钟消耗P1瓦。在汤姆最后一次移动鼠标或触摸键盘后的T1分钟，屏幕保护程序启动，每分钟的功耗变化为P2瓦。最后，从屏幕保护程序启动到T2分钟后，笔记本电脑切换到“睡眠”模式，每分钟消耗P3瓦。 当笔记本电脑处于第二或第三模式时，如果汤姆移动鼠标或触摸键盘，则切换到第一种(正常)模式。 汤姆使用笔记本电脑工作的时间可以分为n个时间间期[l1, r1]、[l2, r2]、...、[ln, rn]。在每个间期，汤姆连续移动鼠标并按下键盘。 在间期之间，汤姆什么都不做。请找出在间期[l1, rn]笔记本电脑的总耗电量。  【输入形式】        第一行包含6个整数n、P1、P2、P3、T1、T2(1<=n<=100，0<=P1、P2、P3<=100，1<=T1、T2<=60)。接下来的n行包含了汤姆工作的期间，第i行是两个用空格分隔的整数li和ri(0<=li<=ri<=1440, 当i<n时ri<li+1）, 表示工作期间的开始时间和结束时间。  【输出形式】        输出总的耗电量。 【样例输入】  2 8 4 2 5 10 20 30 50 100  【样例输出】  570  【样例说明】 【评分标准】 |
| **电话号码** | |

|  |  |
| --- | --- |
| **32.** | 【问题描述】  Vasya有几本电话簿，记录了他的朋友们的电话号码，每一个朋友都可以有一或几个电话号码。  Vasya决定整理关于朋友电话号码的信息。给定n个字符串，来自于Vasya的电话簿中的条目。每一条都以朋友的姓名开头，然后跟着当前条目中的电话号码个数，然后是本人的电话号码。有可能几个相同的电话被记录在同一个记录中。  Vasya还认为，如果电话号码a是电话号码b的后缀（也就是说，号码b以a结尾），这两个号码被当作同一个电话号码，那么a被认为是无城市代码，它不应该被考虑。  输出整理后Vasya朋友的电话号码信息。有可能两个不同的人有相同的号码。如果一个人有两个电话号码x和y，x是y的后缀（即y以x结尾），则不输出x。  如果Vasya的电话簿中的某些朋友记录了几次，那么只需要记录一次。  【输入形式】  输入第一行一个整数n(1<=n<=20)，Vasya的电话簿上的条目数。  以下n行后面是描述中的格式记录。 朋友的姓名中不包含空字符，长度不超过10位，由小写英文字母组成。电话号码个数在1~10之间。每个电话号码的长度范围在1~10之间，可以包含前导0。  【输出形式】  输出Vasya的朋友的电话号码的有序信息。首先输出电话簿中的朋友数目m。  接下来的m行，包含以格式“姓名 电话号码个数 电话号码1 ... 电话号码k"的条目，号码间以空格分隔。每个记录包含当前朋友的所有电话号码。  每个条目输出按照姓名字母序进行排序，电话号码按照从小到大的顺序排列（注意电话号码："1"<"01"、"12"<"012"，依此类推）  【样例输入】  4 ivan 3 123 123 456 ivan 2 456 456 ivan 8 789 3 23 6 56 9 89 2 dasha 2 23 789  【样例输出】  2 dasha 2 23 789  ivan 4 2 123 456 789 |
| **内存管理** | |

|  |  |
| --- | --- |
| **33.** | 【问题描述】         离第一个操作系统HNU-OS发布已经没有多少时间了，但它的一些组件还没有完成，内存管理器就是其中之一。根据开发人员的计划，在第一个版本中，内存管理器将非常简单和直观。它将支持三个操作：   * alloc n —— 分配n个字节内存，返回已分配块的正整数标识符x(x初始值为0，每次分配增长1) * erase x —— 删除标识符x所在的块 * defragment —— 整理空余内存碎片，将所有块尽量靠近内存的开始位置，并保持各自的顺序          在此情况下，内存模型非常简单，它是一个m字节的序列，为了方便起见，从第一个字节到第m字节进行编号。         第一个操作alloc n有一个参数n，表示被分配的内存块大小。在处理此操作时，内存中将分配n个连续字节的空闲块。 如果这些块的数量超过一个，则优先选择最接近内存开始(即第一个字节)的块。 所有这些字节都被标记为非空闲，内存管理器返回一个32位整数数字令牌，代表该块的标识符。 如果不可能分配这样大小的空闲块，则返回NULL。         第二个操作erase x以x为参数，表示某个块的标识符。此操作释放系统内存，将此块的字节标记为空闲以供进一步使用。 如果此标识符没有指向先前分配的块(该块尚未被释放)，则返回ILLEGAL\_ERASE\_ARGUMENT。         最后一个操作defragment没有任何参数，只会使占用的内存部分更接近内存的开始，而不会更改它们各自的顺序。         在当前的实现中，将使用从1开始的连续整数作为标识符。每个成功的alloc操作过程都应该返回接下来的编号。不成功的alloc操作不影响计数。         编写内存管理器的实现，为每个alloc命令输出返回的值，为所有失败的erase命令输出ILLEGAL\_ERASE\_ARGUMENT。  【输入形式】         输入数据的第一行包含两个正整数t和m（1<=t<=500, 1<=m<=105)，其中t表示需要内存管理器来处理的操作个数，m表示有效的内存字节大小。接下来的t行每一行代表一个操作。 【输出形式】         输出有多行，每行或者是alloc操作的结果，或者是失败的erase操作的结果ILLEGAL\_ERASE\_ARGUMENT。其顺序与输入的操作次序一致。  【样例输入】  6 10 alloc 5 alloc 3 erase 1 alloc 6 defragment alloc 6  【样例输出】  1 2 NULL 3 |
| **最大报销额** | |

|  |  |
| --- | --- |
| **34.** | 【问题描述】现有一笔经费可以报销一定额度的发票。允许报销的发票类型包括买图书（A类）、文具（B类）、差旅（C类），要求每张发票的总额不得超过1000元，每张发票上，单项物品的价值不得超过600元。现请你编写程序，在给出的一堆发票中找出可以报销的、不超过给定额度的最大报销额。  【输入形式】测试输入包含若干测试用例。每个测试用例的第1行包含两个正数 Q 和 N，其中 Q 是给定的报销额度，N（N<=30）是发票张数。随后是 N 行输入，每行的格式为：       m Type\_1:price\_1 Type\_2:price\_2 … Type\_m:price\_m       其中正整数 m 是这张发票上所开物品的件数，Type\_i 和 price\_i 是第 i 项物品的种类和价值。物品种类用一个大写英文字母表示。当N为0时，全部输入结束，相应的结果不要输出。  【输出形式】对每个测试用例输出1行，即可以报销的最大数额，精确到小数点后2位。  【样例输入】  200.00 3 2 A:23.50 B:100.00 1 C:650.00 3 A:59.99 A:120.00 X:10.00 1200.00 2 2 B:600.00 A:400.00 1 C:200.50 1200.50 3 2 B:600.00 A:400.00 1 C:200.50 1 A:100.00 100.00 0  【样例输出】  123.50 1000.00 1200.50  【样例说明】 【评分标准】 |
| **占座位** | |

|  |  |
| --- | --- |
| **35.** | 【问题描述】sun所在学校的教室座位每天都是可以预占的。 一个人可以去占多个座位，而且一定是要连续的座位，如果占不到他所要求的这么多座位，那么他就一个座位也不要了。为了降低难度，每次分配座位按座位号从小到大查找，采用最先适配法分配座位。  【输入形式】输入有多组数据。 每组数据输入座位排数n，0<n<=100（座位的排列数相等，座位是按每行从左到右依次排序的,，第1行的最右边一个座位与第二行的第一个座位视为连续座位），m（ 0<m<=min(100,n\*n) ）个人。 然后输入k（0<k<=100），最后输入k个命令。 命令只有两种： 1.in id num（代表id,0<=id<m,要占num个座位，若占不到连续的num(0<num<=20)个座位表示该命令无效） 2.out id（代表id要释放他之前占的所有座位） 注意：如果id之前占过座还没释放那么之后他的in命令都是无效的， 如果id之前没占过座位那么他的out命令也是无效的。  【输出形式】对每个in命令输出yes或者no，如果命令有效则输出yes，无效则输出no。 在yes no后面只带有回车，不带其他任何字符。  【样例输入】  4 10  9  in 1 7  in 2 3  in 3 3  in 3 3  in 4 3  out 2  in 5 6  out 3  in 5 6  【样例输出】  yes yes  yes  no  yes  yes  no  yes  yes |
| **打牌** | |

|  |  |
| --- | --- |
| **36.** | 【问题描述】  牌只有1到9，手里拿着已经排好序的牌a，对方出牌b，用程序判断手中牌是否能够压过对方出牌。  规则：出牌牌型有5种    [1]一张 如4 则5…9可压过  [2]两张 如44 则55，66，77，…，99可压过  [3]三张 如444 规则如[2]  [4]四张 如4444 规则如[2]  [5]五张 牌型只有12345 23456 34567 45678 56789五个，后面的比前面的均大。  【输入形式】  输入有多行，第一行代表手中的牌，长度不超过200个数字。接下来的每一行代表每次对方出的牌。  【输出形式】  输出有多行，代表手中的牌是否能压过对方出的牌，压过输出YES， 并列出所有可选项，可选项之间用空格分隔。 否则输出NO。  【样例输入】  17624234556367  33  222  34567  【样例输出】  YES 44 55 66 77  YES 666  NO |
| **魔咒词典** | |

|  |  |
| --- | --- |
| **37.** | 【问题描述】  哈利波特在魔法学校的必修课之一就是学习魔咒。据说魔法世界有100000种不同的魔咒，哈利很难全部记住，但是为了对抗强敌，他必须在危急时刻能够调用任何一个需要的魔咒，所以他需要你的帮助。  给你一部魔咒词典。当哈利听到一个魔咒时，你的程序必须告诉他那个魔咒的功能；当哈利需要某个功能但不知道该用什么魔咒时，你的程序要替他找到相应的魔咒。如果他要的魔咒不在词典中，就输出“what?”  【输入形式】  首先列出词典中不超过100000条不同的魔咒词条，每条格式为：  [魔咒] 对应功能  其中“魔咒”和“对应功能”分别为长度不超过20和80的字符串，字符串中保证不包含字符“[”和“]”，且“]”和后面的字符串之间有且仅有一个空格。词典最后一行以“@END@”结束，这一行不属于词典中的词条。 词典之后的一行包含非负整数N（0=<N<=1000），随后是N个测试用例。每个测试用例占一行，或者给出“[魔咒]”，或者给出“对应功能”。  【输出形式】  每个测试用例的输出占一行，输出魔咒对应的功能，或者功能对应的魔咒。如果魔咒不在词典中，就输出“what?”  【样例输入】  [expelliarmus] the disarming charm  [rictusempra] send a jet of silver light to hit the enemy  [tarantallegra] control the movement of one's legs  [serpensortia] shoot a snake out of the end of one's wand  [lumos] light the wand  [obliviate] the memory charm  [expecto patronum] send a Patronus to the dementors  [accio] the summoning charm  @END@  4  [lumos]  the summoning charm  [arha]  take me to the sky  【样例输出】  light the wand  accio  what?  what? |
| **小A的计算器** | |

|  |  |
| --- | --- |
| **38.** | 【问题描述】  以往的操作系统内部的数据表示都是二进制方式，小A新写了一个操作系统，系统内部的数据表示为26进制，其中0-25分别由a-z表示。 现在小A要在这个操作系统上实现一个计算器，这个计算器要能实现26进制数的加法运算。你能帮小A实现这个计算器吗？  【输入形式】  输入的第一行包括一个整数N(1<=N<=100)。 接下来的N行每行包括两个26进制数x和y，它们之间用空格隔开，每个数的位数最多为10位,我们可以保证相加的结果的位数最多也是10位。每个数会用小A所设计的操作系统中的表示方法来表示，如：bsadfasdf。即每个数的各个位均由26个小写字母a-z中的一个来表示。  【输出形式】  输出x和y相加后的结果，结果也要用题目中描述的26进制数来表示。  【样例输入】  4  ba cd  c b  b c  ba c  【样例输出】  dd  d  d  bc |
| **导弹防御系统** | |

|  |  |
| --- | --- |
| **39.** | 【问题描述】  某国为了防御敌国的导弹袭击，开发出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭，并观测到导弹依次飞来的高度，请计算这套系统最多能拦截多少导弹。拦截来袭导弹时，必须按来袭导弹袭击的时间顺序，不允许先拦截后面的导弹，再拦截前面的导弹。  【输入形式】  每组输入有两行，  第一行，输入雷达捕捉到的敌国导弹的数量k（k<=25），  第二行，输入k个正整数，表示k枚导弹的高度，按来袭导弹的袭击时间顺序给出，以空格分隔。  【输出形式】  每组输出只有一行，包含一个整数，表示最多能拦截多少枚导弹。  【样例输入】  8  300 207 155 300 299 170 158 65  【样例输出】  6 |
| **愚人节的礼物** | |

|  |  |
| --- | --- |
| **40.** | 【问题描述】  四月一日快到了，Vayko 想了个愚人的好办法——送礼物。嘿嘿，不要想的太好，这礼物可没那么简单，Vayko 为了愚人，准备了一堆盒子，其中只有一个盒子里面装了礼物。盒子里面可以再放零个或者多个盒子。假设放礼物的盒子里不再放其他盒子。用()表示一个盒子，B表示礼物，Vayko 想让你帮她算出愚人指数，即最少需要拆多少个盒子才能拿到礼物。  【输入形式】  本题目包含多组测试，请处理到文件结束。每组测试包含一个长度不大于 1000, 只包含'(',')'和'B'三种字符的字符串，代表 Vayko 设计的礼物透视图。你可以假设，每个透视图画的都是合法的。  【输出形式】  对于每组测试，请在一行里面输出愚人指数。  【样例输入】  ((((B)()))())  (B)  【样例输出】  4  1 |
| **最小的K个数** | |

|  |  |
| --- | --- |
| **41.** | 【问题描述】  输入n个整数，找出其中最小的k（k<=n）个不同数。例如输入4,5,1,6,1,7,3,8这8个数字，则最小的4个数字是1,3,4,5。  【输入形式】  每个测试案例包括2行：  第一行为2个整数n，k(1<=n，k<=200000)，表示数组的长度。  第二行包含n个整数，表示这n个数，数组中的数的范围是[0,1000 000 000]。  【输出形式】  对应每个测试案例，输出最小的k个数，并按从小到大顺序打印(如果不存在k个不同的数，则按照实际数量进行输出)。  【样例输入】  8 4  4 5 1 6 2 7 3 8  【样例输出】  1 2 3 4  【问题提示】  推荐使用STL中的set |
| **绩点计算** | |

|  |  |
| --- | --- |
| **42.** | 【问题描述】  学校对本科生的成绩施行绩点制（GPA）。将学生的实际考分根据不同学科的不同学分按一定的公式进行计算。规定如下：  实际成绩        绩点  90-100          4.0  85-89            3.7  82-84            3.3  78-81            3.0  75-77            2.7  72-74            2.3  68-71            2.0  64-67            1.5  60-63            1.0  60以下            0  1. 一门课程的学分绩点=该课绩点\*该课学分  2. 总评绩点=所有学科绩点之和/所有课程学分之和  现要求你编程求出某人的总评绩点(GPA)  【输入形式】  第一行 总的课程数n  第二行 相应课程的学分（两个学分间用空格隔开）  第三行 对应课程的实际得分  此处输入的所有数字均为整数  【输出形式】  输出有一行，总评绩点，保留两位小数  【样例输入】  5  4 3 4 2 3  91 88 72 69 56  【样例输出】  2.52 |
| **数圈** | |

|  |  |
| --- | --- |
| **43.** | 【问题描述】  以1为中心，用2,3,4, ..., n, ..., n\*n的数字围绕着中心输出数圈， 如若n=4，则  7 8 9 10  6 1 2 11  5 4 3 12  16 15 14 13  【输入形式】  一个整数n(1<=n<=10)  【输出形式】  数圈矩阵  【样例输入】  5  【样例输出】  21 22 23 24 25  20 7 8 9 10  19 6 1 2 11  18 5 4 3 12  17 16 15 14 13 |
| **金币** | |

|  |  |
| --- | --- |
| **44.** | 【问题描述】  国王为他的忠诚的骑士支付金币。在他服役的第一天，骑士收到一枚金币。在接下来2天（第二天和第三天的服务），骑士每天收到2金币。在未来三天（第五，第四，和第六天的服务），骑士每天收到三金币。在未来四天（第七，第八，第九，和第十天的服务），骑士每天收到四金币。这一模式的付款方式将继续下去：在接下来的n天骑士每天将收到n枚金币，而在接接下来的n+1天每天将收到n+1枚金币，这里n是正整数。你的程序将确定在任何给定的天数（从第1天开始）支付给骑士的金币总数。  【输入形式】  输入包含至少一行，但不超过21行。输入的每一行包含一个测试案例的数据，即一个整数（1~10000），代表天数。  【输出形式】  每一行输出对应一个测试用例，由天数和支付给骑士的金币总数量组成，中间用空格分隔。  【样例输入】  10  6  10000  1000  21  22  【样例输出】  10 30  6 14  10000 942820  1000 29820  21 91  22 98 |
| **小丑排序** | |

|  |  |
| --- | --- |
| **45.** | 【问题描述】  你在信天翁马戏团（是的，它是由一群小丑组成）从事管理工作，你刚刚写完一个程序的输出是将他们的姓名按长度为非递减的方式排列，名称列表（使每名至少只要它之前的）。然而，你的老板不喜欢这种输出方式，而是希望输出出现更对称，较短的字符串在顶部和底部，而较长的字符串在中间。他的规则是，每一对名称都是在该列表的相对的两端，并且在该组中的第一个名字总是在列表的顶部。比如在下面的第一个例子中，Bo和Pat是第一对，Jean和Kevin是第二对，等等。  【输入形式】  输入由1到多个字符串集合组成，最后一行为0表示输入结束，每个集合开始于一个整数n，表示该集合字符串的个数，接下来n行由n个字符串按长度非递减的方式排列，每个集合至少包含一个但不超过15个字符串，每个字符串不超过25个字符。  【输出形式】  对于每个集合，第一行输出"set-n", n从1开始，接下来的若干行对应输入每个集合重新排列的结果，如样例所示。  【样例输入】  7  Bo  Pat  Jean  Kevin  Claude  William  Marybeth  6  Jim  Ben  Zoe  Joey  Frederick  Annabelle  5  John  Bill  Fran  Stan  Cece  0  【样例输出】  set-1  Bo  Jean  Claude  Marybeth  William  Kevin  Pat  set-2  Jim  Zoe  Frederick  Annabelle  Joey  Ben  set-3  John  Fran  Cece  Stan  Bill |
| **亲和数** | |

|  |  |
| --- | --- |
| **46.** | 【问题描述】  古希腊数学家毕达哥拉斯在自然数研究中发现，220 的所有真约数(即不是自身的约数)之和为：                 1+2+4+5+10+11+20+22+44+55+110＝284。 而 284 的所有真约数为 1、2、4、71、 142，加起来恰好为 220。人们对这样的数感到很惊奇，并称之为亲和数。一般地讲，如果两个数中任何一个数都是另一个数的真约数之和，则这两个数就是亲和数。 你的任务就编写一个程序，判断给定的两个数是否是亲和数。  【输入形式】  输入若干行数据（大于0），每行一个实例,包含两个整数A,B； 其中 0 <= A,B <= 600000 ;  【输出形式】  对于每个测试实例，如果 A 和 B 是亲和数的话输出 YES，否则输出 NO  【样例输入】  220 284  100 200  【样例输出】  YES  NO |
| **Maya历法** | |

|  |  |
| --- | --- |
| **47.** | 【问题描述】  在学术休假期间，M.A. Ya教授在古老的Maya历法上有一个惊人的发现。从一个古老的令人棘手的信息中，教授发现Maya文明以365天为一年，称为Haab，包含19个月。前18个月每月有20天，月份名字为：pop、no、zip、zotz、tzec、xul、yoxkin、mol、chen、yax、zac、ceh、mac、kankin、muan、pax、koyab、cumhu。每月的天数使用数字来表示，从0~19，而不是用名字。Haab的最后一个月叫做uayet，有5天，表示为0、1、2、3、4。玛雅人认为这个月是不吉利的，法院不开庭，贸易停止了，人们甚至停止清扫地板。         出于宗教的目的，Maya人使用另外一套历法，叫做Tzolkin（冬青年）。一年被分为13个期间，每个期间20天。每天被表示为由数字和日期名表示的数对。使用20个名字：imix、ik、akbal、kan、chicchan、cimi、manik、lamat、muluk、ok、chuen、eb、ben、ix、mem、cib、caban、eznab、canac、ahau，以及13个数字，双循环使用。         请注意，每一天都有一个明确的描述。例如，在年初的日子被描述如下：         1 imix, 2 ik, 3 akbal, 4 kan, 5 chicchan, 6 cimi, 7 manik, 8 lamat, 9 muluk, 10 ok, 11 chuen, 12 eb, 13 ben, 1 ix, 2 mem, 3 cib, 4 caban, 5 eznab, 6 canac, 7 ahau, 在下一个期间开始为 8 imix, 9 ik, 10 akbal . . .         年份（包含Haab和Tzolkin)用数字0、1、...来表示，数字0是世界的开始。因此，第一天表示为：         Haab: 0. pop 0        Tzolkin: 1 imix 0        请帮M.A.Ya教授写一个程序，将Haab日历转换为Tzolkin日历。  【输入形式】         在Haab中日期用以下形式表示：                 NumberOfTheDay. Month Year         输入文件的第一行包含文件中输入日期的数目。接下来的n行包含Haab日历格式的n个日期，年份小于5000。  【输出形式】          Tzolkin日期用一下格式：                 Number NameOfTheDay Year          输出包括n行，按照与输入日期对应的顺序，输出tzolkin日历格式日期。  【样例输入】  3 10.zac 0 0.pop 0 10.zac 1995  【样例输出】  3 chuen 0 1 imix 0 9 cimi 2801  【样例说明】 【评分标准】 |
| **带通配符的数** | |

|  |  |
| --- | --- |
| **48.** | 【问题描述】给定一个可以带通配符问号的正整数W，问号可以代表任意一个一位数字。再给定一个正整数X，和W具有同样的长度。问有多少个整数符合W的形式并且比X大？  【输入形式】多组数据，每组数据两行，第一行是W，第二行是X，它们长度相同，在[1..10]之间。  【输出形式】每行一个整数表示结果。  【样例输入】  36?1?8 236428 8?3 910 ? 5  【样例输出】  100 0 4  【样例说明】 【评分标准】 |
| **487-3279** | |

|  |  |
| --- | --- |
| **49.** | 每个人都喜欢有令人难忘的电话号码。要想让电话号码变得令人难忘的一种方法是拼出一个令人难忘的单词或短语。例如，你可以拨打滑铁卢大学的电话，拨打令人难忘的电话号码TUT-GLOP。         有时只有一部分号码被用来拼写一个单词，例如，你可以拨打310-gino从Gino's订购披萨。         要使电话号码令人难忘的另一种方法是以一种令人难忘的方式对数字进行分组。你可以从比萨饼小屋中订购比萨饼，方法是拨打他们的“3个10”，即号码3-10-10-10。         电话号码的标准格式是七位的十进制数字，第三和第四位之间包含连字符（例如888-1200）。电话的键盘提供字母到数字的映射，如下所示：         A, B, C映射到2         D, E, F映射到3         G, H, I映射到4         J, K, L映射到5         M, N, O映射到6         P, R, S映射到7         T, U, V映射到8         W, X, Y映射到9         Q和Z没有映射。连接符不拨号，必要时可加上或去除。TUT-GLOP的标准格式是888-4567，310-GINO的标准格式是310-4466，3-10-10-10的标准格式是310-1010。         当两个电话号码有相同的标准格式时是等价的（拨同样的号码）。         你的公司正在编制本地企业的电话号码目录，作为质量控制的一部分，你需要检查没有两个（或多个）企业具有相同的电话号码。  【输入形式】  输入包括一个案例。输入的第一行为一个正整数，指定目录中电话号码的数目(最多100，000)。其余的各行列出目录中的电话号码，每个号码单独占一行。每个电话号码都是一个由十进制数字、大写字母(不包括Q和z)和连字符组成的字符串。字符串中的七个字符或是数字或是字母。  【输出形式】  对于出现超过一次的每个号码，按照标准格式每个输出一行，然后是空格，接着输出出现的次数。只出现1次的电话号码不输出。  【样例输入】  12 4873279 ITS-EASY 888-4567 3-10-10-10 888-GLOP TUT-GLOP 967-11-11 310-GINO F101010 888-1200 -4-8-7-3-2-7-9- 487-3279  【样例输出】  310-1010 2 487-3279 4 888-4567 3  【样例说明】 【评分标准】 |
| **缺席考试的是谁？** | |

|  |  |  |
| --- | --- | --- |
| **50.** | 【问题描述】  程序设计考试结束了，传来个不好的消息：有一个学生没参加考试!需要尽快知道缺席考试的人是谁，以便尽快做出处理。  糟糕的是，尽管有签到表，但由于人数较多，签到情况比较混乱：有的签到表签在一张白纸上，有的虽然签在名册上，但并不是签在自己姓名旁，更有学生签到了别的签到表上……  现在只能根据这2n-1个姓名（名册上有n个学生姓名，签到有n-1个姓名，签到姓名和名册姓名可能混在一起了），来找到缺席考试的人是谁。唯一一个有利的条件是所有参加考试的人都签了名，且只签一次，签名也都正确无误。  现在任务交给你：编写一个程序，找出缺席考试的是谁。  【输入形式】  有多组测试数据。  每组测试数据开始一行，是一个正整数n，表示总人数，n=0意味着输入结束并且不需要处理。  以下2n-1行，每行一个字符串，长度不超过20，表示一个人的姓名。姓名有大小写的英文字母、常用汉字组成(注意每个汉字占2个字节，中英文姓名都不排除有重名情况)。  40%的测试数据1 ≤ n≤ 10；  30%的测试数据1 ≤ n≤ 100；  20%的测试数据1 ≤ n≤ 103；  10%的测试数据1 ≤ n≤ 104；  提示：大量输入数据，C/C++输入推荐使用scanf函数  【输出形式】  对于每组测试数据，输出一行，只包含一个字符串，表示缺席的人的姓名。  【样例输入】  2  张三  张三  李四  0  【样例输出】  李四 |  |
|  |  |  |
| **n， 还是n** | | |

|  |  |
| --- | --- |
| **51.** | 【问题描述】  输出 包含n 或者是n的倍数的所有数  【输入形式】  正整数 m,n（0<m，n<1000000）  【输出形式】  从小到大排列的不大于 m 的特殊正整数（包含n，或者是n的倍数）。  【样例输入1】  20 7  【样例输出1】  7 14 17  【样例输入2】  200 11  【样例输出2】  11 22 33 44 55 66 77 88 99 110 111 112 113 114 115 116 117 118 119 121 132 143 154 165 176 187 198  【样例说明】  包含n的数可以考虑使用字符串查找解决 【评分标准】 |
| -**循环数** | |

|  |  |
| --- | --- |
| **52.** | 【问题描述】        循环数是n位长度的整数，当乘以从1到n的任何整数时，产生原始数字的“循环”。也就是说，如果考虑最后一个数字之后的数字“绕”回到第一个数字，两个数字中的数字序列将是相同的，尽管它们可能从不同的位置开始。例如，数字142857是循环的，如下表所示：          142857 \*1 = 142857         142857 \*2 = 285714         142857 \*3 = 428571         142857 \*4 = 571428         142857 \*5 = 714285         142857 \*6 = 857142         编写一个程序来确定数字是否是循环数。  【输入形式】         输入一个数，长度在2到60位之间(请注意，前面的零不应该被删除，它们被认为是确定n的大小和计数的一部分，因此，“01”是一个两位数的数字，与“1”是一个一位数的数字不同。) 。  【输出形式】         对于每个输入，输出一行(Yes或No)标识它是否是循环数。  【样例输入】  142857  【样例输出】  Yes  【样例说明】  【评分标准】 |
| **棋盘** | |

|  |  |
| --- | --- |
| **53.** | 【问题描述】          棋盘是指一个行和列编号从1~N的NxN的二进制矩阵，当行号和列号之和为偶数时该矩阵对应位置为黑色的(1)，否则为白色的(0)。以下图示为N=1、2、3时的棋盘。  http://115.157.200.135/userfiles/image/151123141390500455413.jpg          給出一个NxN的二进制矩阵，请找出位于该矩阵内的最大尺寸的完整棋盘，以及最大尺寸棋盘的数量（棋盘可以交叠）。  【输入形式】         每个测试用例的第一行是一个正整数N(1<=N<=2000)，表示給定矩阵的行数和列数，接下来的N行描述了这个矩阵：每行有N个字符，既可以是“1”（代表黑块），也可以是“0”（代表白块）。矩阵至少包含一个“1”字符。  【输出形式】         输出最大尺寸棋盘的行列的大小，以及最大棋盘的个数，以空格分隔。  【样例输入】  5 00101 11010 00101 01010 11101  【样例输出】  3 3  【样例说明】 【评分标准】 |
| **字符串反转3** | |

|  |  |
| --- | --- |
| **54.** | 【问题描述】         给出一个字符串，请将其每个单词反转后输出。  【输入形式】        输入第一行为一个正整数N，表示测试用例数，接下来的N行，每行一个字符串。  【输出形式】        输出N行，每行对应一个反转后的字符串。  【样例输入】  3 olleh !dlrow m'I morf .unh I ekil .tae  【样例输出】  hello world! I'm from hnu. I like eat. |
| **字符串反转2** | |

|  |  |
| --- | --- |
| **55.** | 【问题描述】           给定一个句子（只包含字母和空格）， 将句子中的单词位置反转，单词用空格分割, 单词之间只有一个空格，前后没有空格。 比如： “hello xiao mi”-> “mi xiao hello”  【输入形式】   输入数据有多组，每组占一行，包含一个句子(句子长度小于1000个字符)  【输出形式】        对于每个测试示例，要求输出句子中单词反转后形成的句子  【样例输入】  hello xiao mi I am a student  【样例输出】  mi xiao hello student a am I  【样例说明】 【评分标准】 |
| **飞行棋** | |

|  |  |
| --- | --- |
| **56.** | 【问题描述】         大家当年一定都下过飞行棋吧。现在Lele和Yueyue要下的棋和这个很相似，只是更简单一点而已。         棋盘由N个格子组成，分别标记为第0格到第N-1格。格子分为两种，一种是普通格子，即表示在该格可以停留。否则是特殊的格子，一旦走到上面，就要根据上面标记的数飞到相应的格子上。如果飞到一个特殊的格子上，则可以继续飞。         除了第0格外，其他格子都只能容纳一个玩家。即一旦A玩家已经在某个格子上，B玩家又走到这里，A玩家则会被踢回第0格，而B玩家留在这个格子上面。         第N-1个格子是终点，一旦一个玩家走到这个格子上，该玩家获胜，游戏结束。         刚刚开始时，两个玩家都站在第0格上，依次扔骰子，根据骰子显示的点数走相应的格子数。比如，玩家在第0格，扔出了5点，则会走到第5个格子上。如果玩家走得超出了棋盘的范围，则要往回走一定的步数。比如，棋盘一共有7(0~6)个格子,玩家在第4格上，扔出了6点，最终他会走到第2格上(4->5->6->5->4->3->2)。         根据观察，骰子扔出来的数也是有规律的。        对于每一盘棋，扔出的第一个点数为 F0=(A\*C+B)%6+1,第二个点数为 F1=(A\*F0+B)%6+1,第三个点数为 F2=(A\*F1+B)%6+1 ....依此类推。         每一盘棋都是由Lele先走，现在就请你当裁判，看谁能获胜。  【输入形式】        本题目包含多组测试，请处理到文件结束。       每组数据占两行。       第一行有4个整数N,A,B,C(含义见题目描述，6<N<200,0<=A,B,C<=2^31)。       第二行有N个字符串，分别表示棋盘上第0个到第N-1个格子的内容。两个字符串之间用一个空格分隔开。        如果字符串为"N",则表示这个格子为普通格子。否则字符串为"GX"(X为0到N-1之间的整数)的形式，其中X表示玩家走到这个格子时，要马上飞到第X个格子。       数据保证第0个和第N-1个格子一定为"N"。  【输出形式】        对于每组数据，在一行内输出结果。       如果Lele能赢这盘棋，则输出"Lele",如果Yueyue赢的话，就输出"Yueyue"。  【样例输入】  7 1 0 6 N G3 N N N N N 7 1 0 6 N G4 N N N N N  【样例输出】  Lele Yueyue  【样例说明】  测试用例保证能有确定结果。 【评分标准】 |
| **xxx定律** | |

|  |  |  |
| --- | --- | --- |
| **57.** | 【问题描述】         对于一个正整数n，如果是偶数，就把n砍掉一半；如果是奇数，把n变成 3\*n+ 1后砍掉一半，直到该数变为1为止。        请计算需要经过几步才能将n变到1，具体可见样例。  【输入形式】         测试包含多个用例，每个用例包含一个整数n,当n为0 时表示输入结束。（1<=n<=10000）  【输出形式】         对于每组测试用例请输出一个数，表示需要经过的步数,每组输出占一行。  【样例输入】  3  2  0  【样例输出】  5  1  【样例说明】  【评分标准】 |  |
|  |  |  |
| **字符串排序** | | |

|  |  |
| --- | --- |
| **58.** | 【问题描述】         定义一个字符串的无序度为所有位置后面的字母比该位置的字母小的总数之和。比如"DAABEC''这个字符串的无序度是5，因为D后面有4个位置比它小（AABC），E后面有1个比它小（C），其它位置后面没有比自己小的。" AACEDGG "的无序度为1（E后面有一个D比它小）。" ZWQM "的无序度为6，每个位置后面所有的字母都比它小。        现在你的任务是给定一些字符串（只由大写字母组成），把他们按照无序度从小到大排序，如果无序度一样，那么就按照输入的相对顺序排序。  【输入形式】      单组测试数据。      第一行有两个整数n(0 < n <= 50)和m (0 < m <= 100)，分别表示输入的字符串的长度和字符串的个数。      接下来m行，每一行包含一个长度为n的字符串，只由大写字母组成。  【输出形式】     输出m行，表示排序之后的字符串。  【样例输入】  10 6 AACATGAAGG TTTTGGCCAA TTTGGCCAAA GATCAGATTT CCCGGGGGGA ATCGATGCAT  【样例输出】  CCCGGGGGGA AACATGAAGG GATCAGATTT ATCGATGCAT TTTTGGCCAA TTTGGCCAAA  【样例说明】 【评分标准】 |
| **拼写检查** | |

|  |  |  |
| --- | --- | --- |
| **59.** | 【问题描述】         作为一个新的拼写检查程序开发团队的成员，您将编写一个模块，用已知的所有形式正确的词典来检查给定单词的正确性。        如果字典中没有这个词，那么可以用下列操作中的一个来替换正确的单词（从字典中）：        1. 从单词中删除一个字母；        2. 用一个任意字母替换单词中的一个字母；        3. 在单词中插入一个任意字母。        你的任务是编写一个程序，为每个给定的单词找到字典中所有可能的替换。  【输入形式】         输入的第一部分包含所有字典中的词，每个单词占用一行，以一个单一字符“#”作为结束。所有单词都不相同，字典中至多1000个单词。         接下来的部分包含所有需要进行检查的单词，同样每个单词占用一行。这部分也以一个单一字符“#”作为结束。至多有50个单词需要检查。         在输入中所有的单词（字典中的和需要检查的）都仅由小写字母组成，每个最多包含15个字符。  【输出形式】         对于每个在输入中出现的单词，按照它们在输入的第二部分出现的顺序输出一行。如果该单词是正确的（也就是说它包含在字典中）则输出信息：“is correct”；如果该单词不正确，则首先输出该单词，然后输入符号':'（冒号），之后空一格，写出它所有可能的替代，以空格分隔。这些替代的单词按照它们在字典中（输入的第一部分）出现的顺序写出。如果没有可替代的单词，则在冒号后面直接输出换行。  【样例输入】  i is has have be my more contest me too if award # me aware m contest hav oo or i fi mre #  【样例输出】  me is correct aware: award m: i my me contest is correct hav: has have oo: too or: i is correct fi: i mre: more me  【样例说明】 【评分标准】 |  |
|  |  |  |
| **点球大战** | | |

|  |  |
| --- | --- |
| **60.** | 【问题描述】在足球比赛中，有不少赛事，例如世界杯淘汰赛和欧洲冠军联赛淘汰赛中，当比赛双方经过正规比赛和加时赛之后仍然不分胜负时，需要进行点球大战来决定谁能够获得最终的胜利。点球大战的规则非常简单，两方轮流派出球员罚点球，每方各罚5个。当5轮点球结束以后如果仍然不分胜负，则进入一轮定胜负的阶段。两方各派一名球员罚点球，直到有一方罚进而另一方没有进为止。  在北美职业冰球联赛中，也有点球大战。与足球的规则不同的是，它只先罚3轮点球，随后就进入一轮定胜负的阶段，而其他的规则完全一样。  在本题中，输入将给出每次点球是否罚进，而你的任务则是输出一个“比分板”。  【输入形式】输入包含多组数据。每组数据的第一行包含一个整数N(1<=N<=18)，表示双方总共罚了多少个点球，N=0表示输入结束。随后有N行，每行是一个如下形式的字符串：  XXXX good：表示这个点球罚进  或者XXXX no good：表示这个点球没有罚进  其中XXXX表示球员名字（全部由字母和空格组成，保证不会出现歧义）  每一行保证不超过100个字符。  XXXX和good以及XXXX和no、no和good之间保证有且只有1个空格。  good、no good都是小写。本题是大小写相关的。  数据不保证点球大战一定结束，也不保证在结束以后立即结束这组数据（即：不用判断点球大战是否结束，只用把罚进的点球往比分上加即可）。  【输出形式】对每组数据，输出一个比分板。一个点球如果罚进，则在对应的地方标上’O’，如果没有进则标上’X’。先罚球的队伍的信息在上面，后罚的在下面。最右边标上两队的比分。具体格式参考样例输出。注意如果一轮点球只罚了一个，则后面那个点球对应的地方写上’-’。  【样例输入】  6 Riise good Ballack good Gerrard no good Lampard no good Fernando Torres good Malouda good 9 Christiano Ronaldo no good Messi no good Giggs good Abidal no good Carrick good Ronaldinho good Rooney good Henry no good Tevez good 0  【样例输出】  1 2 3 Score O X O 2 O X O 2 1 2 3 4 5 Score X O O O O 4 X X O X - 1  【样例说明】 【评分标准】 |
| -**Engine-字符串** | |

|  |  |  |
| --- | --- | --- |
| **61.** | 【问题描述】谷歌、百度等搜索引擎已经成为了互连网中不可或缺的一部分。在本题中，你的任务也是设计一个搜索论文的搜索引擎，当然，本题的要求比起实际的需求要少了许多。 本题的输入将首先给出一系列的论文，对于每篇论文首先给出标题，然后给出它被引用的次数。然后会有一系列的搜索询问，询问标题中包含特定关键词的论文有哪些。 每一个询问可能包含多个关键词，你需要找出标题包含所有关键词的论文。 “包含”必须是标题中有一个词正好是给定的关键词，不区分大小写。 对每个询问，都按被引用的次数从多到少输出满足条件的论文的标题。如果有被引用的次数相同的论文，则按照论文在输入中的顺序排列，先给出的论文排在前面。  【输入形式】输入包含多组数据。 每组数据首先有一行包含一个整数N(1<=N<=1000)，表示论文的数目，N=0表示输入结束。每组论文的信息第一行是论文的标题，由字母（大小写均可）和空格组成，不超过10个词，每个词不超过20个字符，标题总共不超过250个字符。第二行是一个整数K(0<=K&lt;=108)，表示它被引用的次数。在论文信息结束以后，有一行包含一个整数M(1<=M<=100)，表示询问的数目。接下来有M行，每行是一个询问，由L(1<=L<=10)个空格分开的词构成，每个词不超过20个字符。  【输出形式】对每个询问，按照题目给定的顺序输出满足条件的论文的标题；如果没有满足条件的论文，就不输出。在每组询问的输出之后输出一行“\*\*\*”，在每组数据的输出之后输出一行“---”。  【样例输入1】  6 Finding the Shortest Path 120 Finding the k Shortest Path 80 Find Augmenting Path in General Graph 80 Matching in Bipartite Graph 200 Finding kth Shortest Path 50 Graph Theory and its Applications 40 6 shortest path k shortest path graph path find application 0  【样例输出1】  Finding the Shortest Path Finding the k Shortest Path Finding kth Shortest Path \*\*\* Finding the k Shortest Path \*\*\* Matching in Bipartite Graph Find Augmenting Path in General Graph Graph Theory and its Applications \*\*\* Finding the Shortest Path Finding the k Shortest Path  Find Augmenting Path in General Graph Finding kth Shortest Path \*\*\* Find Augmenting Path in General Graph \*\*\*  \*\*\* ---  【样例输入2】  1 Finding the Shortest Path 120 2 Path  Pat 0  【样例输出2】  Finding the Shortest Path  \*\*\*  \*\*\*  ---  【样例说明】 【评分标准】 |  |
|  |  |  |
| **Web导航** | | |

|  |  |
| --- | --- |
| **62.** | 【问题描述】  标准的Web浏览器具有在最近访问的页面中前后移动的特性。实现这些特性的一种方法是使用两个堆栈来跟踪可以通过前后移动到达的页面。在这个问题中，我们要求实现这一点。         需要支持以下命令：         BACK：将当前页面压入前向堆栈的顶部；从后向堆栈的顶部弹出该页，使其成为新的当前页。如果后向堆栈为空，则该指令忽略。         FORWARD：将当前页面压入后向堆栈的顶部；从前向堆栈的顶部弹出该页，使其成为新的当前页。如果前向堆栈为空，则该指令忽略。         VISIT：将当前页面压入后向堆栈的顶部，将URL指定为新的当前页。前向堆栈被清空。         QUIT：退出浏览器。         假设浏览器最初在网址http://www.game.org/上加载网页。  【输入形式】输入是一个命令序列。命令关键字BACK、FORWARD、VISIT和QUIT都是大写。URL中无空格，最多有70个字符。假定在任何时候，每个堆栈中没有问题实例需要超过100个元素。输入的结尾由QUIT命令标识。  【输出形式】除QUIT外的每个命令，如果命令没有被忽略，则在命令执行后输出当前页面的URL，否则，打印"Ignored"。每个命令的输出独立打印一行。QUIT命令无输出。  【样例输入】  VISIT http://game.ashland.edu/ VISIT http://game.baylor.edu/acmicpc/ BACK BACK BACK FORWARD VISIT http://www.our.com/ BACK BACK FORWARD FORWARD FORWARD QUIT  【样例输出】  http://game.ashland.edu/  http://game.baylor.edu/acmicpc/  http://game.ashland.edu/  http://www.game.org/  Ignored  http://game.ashland.edu/  http://www.our.com/  http://game.ashland.edu/  http://www.game.org/  http://game.ashland.edu/  http://www.our.com/  Ignored  【样例说明】 【评分标准】 |
| **数的距离差** | |

|  |  |
| --- | --- |
| **63.** | 【问题描述】  给定一组正整数，其中最大值和最小值分别为Max和Min, 其中一个数x到Max和Min的距离差定义为：        abs(abs(x-Max)-(x-Min))  其中abs()为求一个数的绝对值  【输入形式】  包括两行，第一行一个数n，表示第二行有n个正整数  【输出形式】  输出一个数x，该数在所有n个数中的距离差最小；如果有两个数的距离差都是最小，输出较小的哪个  【样例输入1】  5  3 1 7 5 9  【样例输出1】  5  【样例输入2】  3  1 3 2  【样例输出2】  2 |