

评测题

2022. 2. 13上午 8:10-11:10



预祝各位虐场成功!

注意:

1. 评测在 windows 下，行未必加回车表示输出完成。
2. **仔细读题，不要跳过任何一个字**，坠吼先通读一遍 pdf。
3. 不准喊“这套题好难啊”，影响别人 AK。
4. 可以喊“这套题好水啊”“我要 ak 辣”，不过 ak 之后不要和别人谈笑风生，影响他人爆零。
5. 题目的评测要求：**时限：1秒，内存：256M。**

假币问题

(coin.cpp/.c)

赛利有12枚银币。其中有11枚真币和1枚假币。假币看起来和真币没有区别，但是重量不同。但赛利不知道假币比真币轻还是重。于是他向朋友借了一架天平。朋友希望赛利称三次就能找出假币并且确定假币是轻是重。例如:如果赛利用天平称两枚硬币，发现天平平衡，说明两枚都是真的。如果赛利用一枚真币与另一枚银币比较，发现它比真币轻或重，说明它是假币。经过精心安排每次的称量，赛利保证在称三次后确定假币。

Input (coin.in)

第一行有一个数字n，表示有n组测试用例。
对于每组测试用例：
输入有三行，每行表示一次称量的结果。赛利事先将银币标号为A-L。每次称量的结果用三个以空格隔开的字符串表示：天平左边放置的硬币、天平右边放置的硬币、平衡状态。其中平衡状态用`up`,`down`,或`even`表示, 分别为右端高、右端低和平衡。天平左右的硬币数总是相等的。

Output (coin.out)

输出哪一个标号的银币是假币，并说明它比真币轻还是重(heavy or light)。

Sample

Input	Output
1 ABCD EFGH even ABCI EFJK up ABIJ EFGH even	K is the counterfeit coin and it is light.

生理周期

(cycle.cpp/.c)

人生来就有三个生理周期，分别为体力、感情和智力周期，它们的周期长度为23天、28天和33天。每一个周期中有一天是高峰。在高峰这天，人会在相应的方面表现出色。例如，智力周期的高峰，人会思维敏捷，精力容易高度集中。因为三个周期的周长不同，所以通常三个周期的高峰不会落在同一天。对于每个人，我们想知道何时三个高峰落在同一天。对于每个周期，我们会给出从当前年份的第一天开始，到出现高峰的天数（不一定是第一次高峰出现的时间）。你的任务是给定一个从当年第一天开始数的天数，输出从给定时间开始（不包括给定时间）下一次三个高峰落在同一天的时间（距给定时间的天数）。例如：给定时间为10，下次出现三个高峰同天的时间是12，则输出2（注意这里不是3）。

Input (cycle.in)

一行，包含四个整数：p, e, i和d，相邻两个整数之间用单个空格隔开。 p, e, i分别表示体力、情感和智力高峰出现的时间（时间从当年的第一天开始计算）。d 是给定的时间，可能小于p, e, 或 i。 所有给定时间是非负的并且小于等于365, 所求的时间小于等于21252。

Output (cycle.out)

一个整数，即从给定时间起，下一次三个高峰同天的时间（距离给定时间的天数）。

Sample

Input	Output
4 5 6 7	16994

完美立方
(cube.cpp/.c)

形如 $a^3 = b^3 + c^3 + d^3$ 的等式被称为完美立方等式。例如 $12^3 = 6^3 + 8^3 + 10^3$ 。编写一个程序，对任给的正整数N ($N \leq 100$)，寻找所有的四元组(a, b, c, d)，使得 $a^3 = b^3 + c^3 + d^3$ ，其中a,b,c,d 大于 1, 小于等于N，且 $b \leq c \leq d$ 。

Input (cube.in)

一个正整数N ($N \leq 100$)。

Output (cube.out)

每行输出一个完美立方。输出格式为：
Cube = a, Triple = (b,c,d)
其中a,b,c,d所在位置分别用实际求出四元组值代入。

请按照a的值，从小到大依次输出。当两个完美立方等式中a的值相同，则b值小的优先输出、仍相同则c值小的优先输出、再相同则d值小的先输出。

Sample

Input	Output
24	Cube = 6, Triple = (3,4,5) Cube = 12, Triple = (6,8,10) Cube = 18, Triple = (2,12,16) Cube = 18, Triple = (9,12,15) Cube = 19, Triple = (3,10,18) Cube = 20, Triple = (7,14,17) Cube = 24, Triple = (12,16,20)

组合的输出

([compages.cpp/c/pas](#))

【问题描述】

排列与组合是常用的数学方法，其中组合就是从 n 个元素中抽出 r 个元素(不分顺序且 $r \leq n$)，我们可以简单地将 n 个元素理解为自然数 $1, 2, \dots, n$ ，从中任取 r 个数。现要求你用递归的方法输出所有组合。

例如 $n = 5, r = 3$ ，所有组合为：

1 2 3 1 2 4 1 2 5 1 3 4 1 3 5 1 4 5 2 3 4 2 3 5 2 4 5 3 4 5

【输入格式】

一行两个自然数 n, r ($1 < n < 21, 1 \leq r \leq n$)。

【输出格式】

所有的组合，每一个组合占一行且其中的元素按由小到大的顺序排列，相邻元素用一个空格分隔，所有的组合也按字典顺序。

【样例输入】 ([compages.in](#))

5 3

【样例输出】 ([compages.out](#))

1 2 3

1 2 4

1 2 5

1 3 4

1 3 5

1 4 5

2 3 4

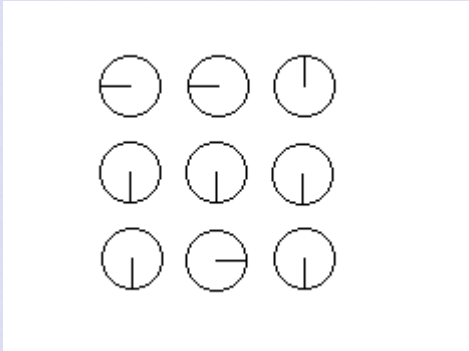
2 3 5

2 4 5

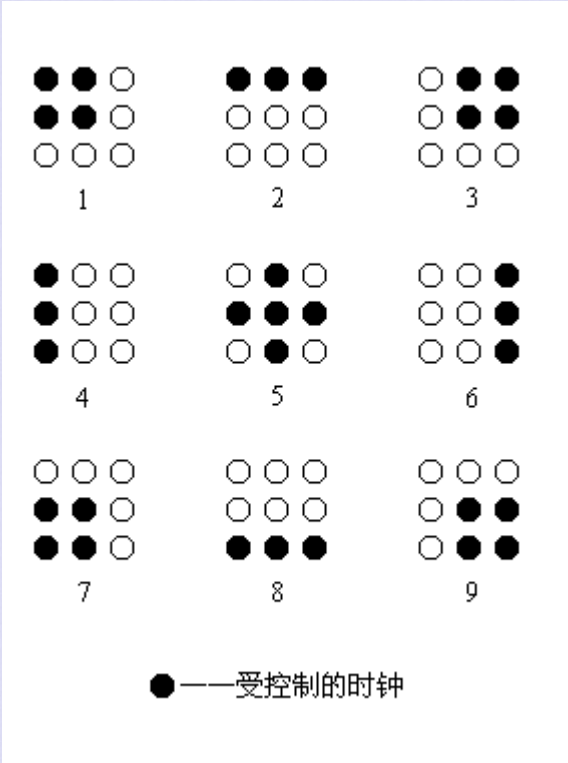
3 4 5

拨钟问题
clock.c/.cpp

在下图（1）所示的3*3矩阵中有9个时钟，我们的目标是旋转时钟指针，使所有时钟的指针都指向12点。允许旋转时钟指针的方法有9种，每一种移动用一个数字号（1，2，...，9）表示。图（2）表示9个数字号与相应的受控制的时钟，这些时钟在图中以黑色小圆标出，其指针将顺时针旋转90度。



图(1) 九种时钟状态



图(2) 九种被控制方式

现在需要用最少的移动，将9个时钟的指针都拨到12点的位置。

Input [clock.in](#)

9个整数，表示各时钟指针的起始位置，相邻两个整数之间用单个空格隔开。其中，0=12点、1=3点、2=6点、3=9点。

Output [clock.out](#)

输出一个最短的移动序列，使得9个时钟的指针都指向12点。按照移动的序号从小到大输出结果。相邻两个整数之间用单个空格隔开。若无解，则输出"No answer."，不输出双引号。

Sample

Input	Output
3 3 0 2 2 2 2 1 2	4 5 8 9

具体的移动方案如下图所示，5 8 4 9 升序排列后即为 4 5 8 9。

