

2005 年合肥市青少年信息学 (计算机) 竞赛中学组试题

注意事项

1. 务必看清题目，严格按照所要求的格式输入、输出。
2. 在调试程序时请先使用题目中的示例数据，然后再自行设计多组测试数据进行调试。
3. 测试有严格的时间限制，请尽可能优化算法。
4. 命名规则：
 - (1) 每题都规定了该题的英文名称。
 - (2) 程序文件和数据文件的主文件名都是该题的英文名字。
 - (3) 程序文件扩展名采用语言环境的默认扩展名。
 - (4) 数据文件都是文本文件，输入和输出文件的扩展名分别是 `.in` 和 `.out`。
5. 程序应从输入文件读取数据，并严格地按照规定的输出格式将结果输出到输出文件中。输入数据文件和输出数据文件都与程序在同一个目录中，由于程序所在目录是不确定的，因此不允许在文件名中含有盘符信息和任何形式的路径信息。
6. 选手在竞赛结束时应在软磁盘上提交所完成各题的各类文件，每位选手必须在所提交的软盘根目录下建立以本人参赛号命名的文件夹。选手在竞赛结束时应将所完成各题的各类文件，包括源程序文件和编译所产生的可执行文件 (即扩展名为 `.exe` 的文件) 拷入该文件夹下，并在硬盘上留下备份。

题目

1. 双色球彩票 (*lottery*)

每个星期小雪都会拿出一部分自己的零花钱买双色球彩票，一来是为了支持国家的彩票发展，其次希望通过中奖来实现自己的愿望。"双色球"每注投注号码由 6 个红色球号码和 1 个蓝色球号码组成。红色球号码从 1-33 中选择；蓝色球号码从 1-16 中选择。中奖规定为：

一等奖 : 7 个号码相符(6 个红色球号码和 1 个蓝色球号码)(红色球号码顺序不限 , 下同) ;

二等奖 : 6 个红色球号码相符 ;

三等奖 : 5 个红色球号码和 1 个蓝色球号码相符 ;

四等奖 : 5 个红色球号码或 4 个红色球号码和 1 个蓝色球号码相符 ;

五等奖 : 4 个红色球号码或 3 个红色球号码和 1 个蓝色球号码相符 ;

六等奖 : 1 个蓝色球号码相符(有无红色球号码相符均可)。

并且规定 , 每张彩票只能中其能中的最高的奖项 , 不计复奖。

现在已知中奖号码和小庆买的若干张彩票的号码 , 请你写一个程序帮助小庆判断他买的彩票的中奖情况。

输入 : 输入文件中第一行只有一个自然数 n ($n < 1000$) , 表明小庆买的彩票张数 ; 第二行存放了 7 个经公证处摇出的中奖球号码 , 每个号码之间用四个空格隔开 , 前 6 个号码代表中奖红球的号码 , 第 7 个号码代表中奖蓝色球的号码 ; 随后的 n 行表示小庆买的 n 张彩票的号码 , 每行的前 6 个数都代表红球的号码 ; 第 7 个数代表蓝色球的号码 , 每个数之间也用四个空格隔开。

输出 : 单行一次输出小庆一等奖至六等奖的中奖张数 , 中间用一个空格隔开。

样例 :

输入 : (lottery.in)

```
2
9      11      2      25      31      22      14
2      11      9      22      31      25      9
11      1      9      2      31      14      12
```

输出 : (lottery.out)

0 1 0 0 1 0

2. 法师的农场 (*Cultivating*)

大法师突然想过一种凡人的生活 , 于是他隐居在一个偏远的农场中。然而 , 他并不需要耕地 , 因为借助他的魔法 , 耕地这件事情将变得十分简单。

大法师的农场呈一个三角形 , 每天早晨 , 他离开家 , 来到三角形的一个顶点 , 做法之后再回到家休息 ; 中午 , 精力恢复的法师又来到三角形的另一个顶点做法 , 之后他回到家休息一下午后 , 在傍晚来到第三个顶点做法 , 此时当他到家的时候 , 他只需要念一句咒语 :

abracadabra !!

农场的三个顶点将会同时施展法力 , 瞬间遍布整个农场。无边的法力将为农场内的土地耕耘、施肥、降雨 , 农作物茁壮成长。

然而 , 法师虽然法力无边 , 但却十分不喜欢走路。正因为如此 , 他希望在农场内找到一个位置作为他的家 , 这样他能走尽量少的路 , 完成他每天的农耕工作。

输入 : 输入文件仅有一行 6 个整数 , 两两一组 , 描述农场的 3 个顶点坐标 (输入坐标范围绝对值不大于 10^5)。

输出 : 向输出文件输出一行两个尽量精确的数值 , 表示法师家的位置 (精确到小数点后十位)。

样例 :

输入 (Cultivating.in):

0 0 0 1 0 2

输出 (Cultivating.out):

0 1

注意 : 你的答案需要和标准答案误差的绝对值不超过 10^{-8} 。

3. 学习计划 (plan)

哈利波特终于离开了可怕的舅舅家 , 来到了 Hogwarts 的魔法学校学习魔法。魔法的世界是多么的神奇 , 小哈利对一切都充满了好奇。他想尽可能多的学习魔法。现在的问题是有些魔法课的时间有冲突 , 哈利无法在一天内上所有的魔法课。所以需要写一个程序来帮助哈利波特制定一个学习计划 , 来安排第一天的学习 , 使得他能尽可能地上更多的魔法课。注意 , 上课的时间是不能改变的。而且上课的时候不能迟到也不能早退 , 否则魔法老师会对哈利产生不满。可以假设从一个教室到另一个教室的时间短得忽略不计。另外 , 在 Hogwarts 的魔法世界里 , 是不使用 24 小时制的计时方法的 , 它只是简单的使用一个整数来表示当时的时间。

输入 : 每个测试数据开头是一个整数 n ($1 \leq n \leq 1000$), 表示魔法课的总数。接下来 n 行 , 每行包括两个正整数 s 、 t , 分别表示该魔法课的上课时间和下课时间。其中 $s < t$ 。

输出 : 对于每个测试数据 , 在单独一行内输出哈利所能上的最多魔法课数。

样例 :

输入 (plan.in) :

3

1 15

2 19

15 17

输出 (plan.out) :

2

4. 史莱克 (shrek)

怪物史莱克在驴子的帮助下 , 从喷火龙和暴君手中救出了菲奥娜公主。他也有幸将要成为公主的丈夫。在婚礼前 , 他偶然得到了一张地图 , 上面记录了一批宝藏的埋藏地。史莱克很高兴 , 他想找到这批宝藏 , 来为自己和菲奥娜公主办一个盛大的婚礼 , 但是他无法确定是否能到达宝藏的埋藏处 , 而且能够安全返回。他可不想以自己的生命为代价。所以想请你帮忙编写一个程序告诉他结果 , 如果你得出能安全到达和返回 , 史莱克就开始他的寻宝旅程 , 如果答案是否定的 , 旅

程就只好放弃了。

这个地图由一个一个 $20\text{km} \times 20\text{km}$ 的区域组成。史莱克一天恰好也只能走 20km ，也就是说在一天之内，他只能从一个区域走到与其上相邻、下相邻、左相邻或者右相邻的区域。由于一些原因，史莱克只能沿着水平或垂直的方向行走。也就是说他不能沿着左上、右上、左下、右下这样的方向前进。

地图上共有 6 种地形，分别是：沙漠、绿洲、森林、城镇、大路和沼泽。这个地图上也标出了，在各个地形上行走的耗水量。当走在大路上时，一天需要消耗 2kg 的水，在森林里，一天只要消耗 1kg 的水，因为树木会遮挡阳光。当穿越绿洲时，根本就不需要消耗水，因为在绿洲里，如果渴了，随时都能够发现水。如果穿越沙漠，那么一天的耗水量就要上升到 4kg 。在旅途中到达一个城镇是非常幸运的事情，因为，城镇里好客的人们不但会款待史莱克，而且还能够为他最大限度的补充水。但是，如果遇到沼泽地，最好还是小心，因为那里每走一步都要面临死亡的威胁，史莱克可不想冒这样的险，所以，遇到沼泽就意味着无法穿越。在旅途中，如果所带的水没有了，史莱克就不能继续他的旅程了。

在地图上用 S 表示出发点，T 表示宝藏埋藏点，D 表示沙漠，O 表示绿洲，F 表示森林，C 表示城镇，R 表示大路，M 表示沼泽。所以地图可能是下面这个样子：

```
RRRRRRRRRRRRRRRR
RRSRRRFFFFFFFFF
DDDDDDDDFFFFFFFF
DDDDODDDFFFFFFFF
DDDDDDCDFFFFFFFF
DTDDOOOFFFFFFFF
RRRRRRRRRRRRMM
```

对于出发点和宝藏埋藏点，它们被看成和大路一样，也就说出发时要消耗 2kg 的水，返回的时候，也要在宝藏埋藏点消耗 2kg 的水。挖掘宝藏时，消耗的时间是很少的，所以不会消耗水。

史莱克会告诉你他最多能带多少水。现在就要求你，给他一个结果，到底能不能开始他的旅程。如果可以开始的话，你要告诉他当他回到出发点的时候，他最多还能剩余多少的水。因为离他的婚礼还有充足的时间，所以他关心最后最多能剩余多少水超过旅程所花的时间。

注意：如果在旅途中史莱克所带的水降为 0，如果这时候他在城镇里，他依然可以继续他的旅程，因为在那他可以得到足够的补给；如果他在绿洲里，他可以继续走进相邻的绿洲（如果有相邻的绿洲），因为在绿洲里行走不需要消耗水；

如果他在别的地形中，他只能放弃他的旅程了。

输入：第一行有一个非负整数 W ，表示史莱克所能带的最大水量。下面一行有两个整数 m 、 n ，标明了地图的尺寸， m 表示一行有多少个 $20\text{km} \times 20\text{km}$ 的区域， n 表示有多少行。紧接着 n 行，每行有 m 个字符，描述了地图的地形，里面也包括了一个 S (出发点) 和一个 T (宝藏埋藏处)。

输出：如果史莱克能够到达宝藏埋藏点，并且能够安全返回，那么，首先在第一行输出 Yes, 然后换行输出最后最多能剩余的水量。如果不能开始旅程，那么就输出 No。(注意 Yes 和 No，首字母要大写)

样例：

输入 (sherk.in)：

100

15 7

RRRRRRRRRRRRRRRR

RRSRRRFFFFFFFFF

DDDDDDDDFFFFFFFF

DDDODDDFFFFFFFF

DDDDDDCDFFFFFFFF

DTDDOOOFFFFFFFF

RRRRRRRRRRRRRMM

输出 (sherk.out)：

Yes

88