

# 本节涉及计算几何,和 DP

## Translate:USACO/fence4

**Closed Fences** 闭合的栅栏

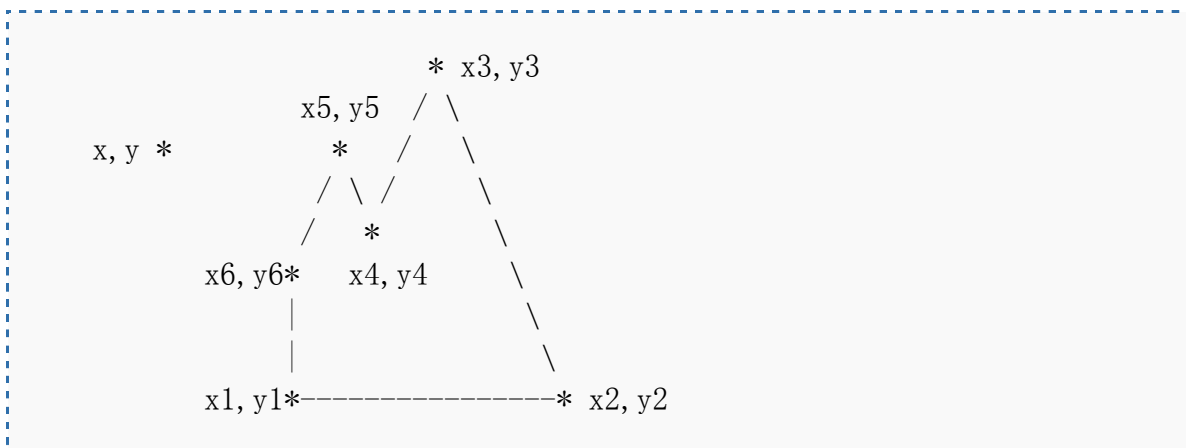
译 by Jeru

### 描述

一个闭合的栅栏是平面上的一些不相交的首尾相连的线段形成的多边形，有  $N$  个角(顶点) ( $3 < N < 200$ )。 顶点不重合，它以逆时针方式以数组  $\{x_i, y_i\}$  给出 ( $i=1,2,\dots,N$ )。

每一对相邻的顶点都是一条栅栏。因此共有  $N$  条栅栏 (定义  $x_{N+1}=x_1, y_{N+1}=y_1$ )。

这里有一个栅栏的例子和一个点  $x,y$ :



请编写一个程序实现下面的任务:

检查输入的顶点列表  $\{x_i, y_i\}$ ,  $i=1,2,\dots,N$ , 判断它是否为一个合法的闭合栅栏。

找出所有可以被站在点 $(x,y)$ 处的人所能看到的栅栏(忽略人的高度)，因为有的栅栏会被另外的栅栏所挡住。

只有当存在从 $(x,y)$ 发射的一条射线第一个穿过栅栏 $i$ 时，栅栏 $i$ 是可以被看见的。如果栅栏是平行于目光的，它并不认为是可以看见的。在上面的例子里，线段 $[x3,y3]-[x4,y4]$ ,  $[x5,y5]-[x6,y6]$ ,  $[x6,y6]-[x1,y1]$ 是可以被 $(x,y)$ 看见的。

## 格式

---

**PROGRAM NAME:** fence4

**INPUT FORMAT:**

(file fence4.in)

第一行:  $N$ , 表示闭合栅栏的顶点数。

第二行: 两个整数  $x$  和  $y$ , 表示观测者的位置。两个整数都是 16 位的。即  $2^{16}$ , 在 longlong 或 longint 范围内。

第 3 到  $N+2$  行: 每行一对整数 $(x,y)$ 表示对应闭合栅栏的第  $k$  个顶点的坐标。坐标以逆时针顺序给出。整数绝对值不超过 1000。

注意: 我添加了该题的新的第 12 个测试点。如果你认为这个点的数据是错误的, 发送邮件到 Rob ([kolstad@ace.delos.com](mailto:kolstad@ace.delos.com)) 在您的邮件主题中一定要包括 **USACO!**

**OUTPUT FORMAT:**

(file fence4.out)

如果给出的序列不是一个合法的闭合栅栏, 那么输出文件只需要输出“NOFENCE”。

栅栏用两端的顶点表示，顶点的输出顺序以输入文件中的顺序为准。把栅栏按照最后一个点在输入文件中的顺序排序。如果两条栅栏的最后一个点是一样的，就以它们第一个点的顺序排序。

## SAMPLE INPUT

```
13
5 5
0 0
7 0
5 2
7 5
5 7
3 5
4 9
1 8
2 5
0 9
-2 7
0 3
-3 1
```

## SAMPLE OUTPUT

```
7
0 0 7 0
5 2 7 5
7 5 5 7
5 7 3 5
-2 7 0 3
0 0 -3 1
0 3 -3 1
```

### 分析

第一个计算几何问题,计算几何问题必须要"踏踏实实"的来,但是也有优化.对于本题目,我使用的是2分法.(我只想到了这个方法,听说还有强大的类似线段树的方法).首先第一问:使用叉积判断任意两条线段是否绝对相交,若有则输出"NOFENCE"并结束.重点是第二问.观察下图.

我们可以很清楚的知道,如果一条线段可以被观察到,他的视野一定不为 0,因此,我们只要先找到待观察线段的中点,并与观察点连线就能得到一条新的线段(中位线).并用新的线段与其它所有线段进行绝对相交判断.同时还要考虑非绝对相交,被点挡住的情况.如果有中位线满足不与所有点相交,那么他一定可以被看到.否则 2 分.边界条件是两个端点挨在一起了,说明没有满足.这时就要考虑控制精度了.一旦精度过高就会超时.反之就会漏解.(0.005 就不错(^o^))具体的实现请看程序。

## Translate:USACO/heritage

**American Heritage** 美国血统

译 By TinyTony

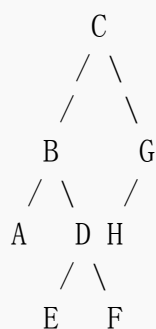
### 描述

农夫约翰非常认真地对待他的奶牛们的血统。然而他不是一个真正优秀的记帐员。他把他的奶牛们的家谱作成二叉树，并且把二叉树以更线性的“树的中序遍历”和“树的前序遍历”的符号加以记录而不是用图形的方法。

你的任务是在被给予奶牛家谱的“树中序遍历”和“树前序遍历”的符号后，创建奶牛家谱的“树的后序遍历”的符号。每一头奶牛的姓名被译为一个唯一的字母。

（你可能已经知道你可以在知道树的两种遍历以后可以经常地重建这棵树。）

显然，这里的树不会有多于 26 个的顶点。这是在样例输入和样例输出中的树的图形表达方式：



树的中序遍历是按照左子树，根，右子树的顺序访问节点。

树的前序遍历是按照根，左子树，右子树的顺序访问节点。

树的后序遍历是按照左子树，右子树，根的顺序访问节点。

## 格式

---

**PROGRAM NAME:** heritage

**INPUT FORMAT:**

(file heritage.in)

第一行： 树的中序遍历

第二行： 同样的树的前序遍历

**OUTPUT FORMAT:**

(file heritage.out)

单独的一行表示该树的后序遍历。

## SAMPLE INPUT

---

```
ABEDFCHG
CBADEFGH
```

## SAMPLE OUTPUT

---

```
AEFDBHGC
```

### 分析:

采取递归的方法建立二叉树。首先取前序遍历的首元素当作二叉树的根，当前元素为根。把前序遍历中的当前元素当作中序遍历的分割点，中序遍历分割点前面的元素一定在当前元素的左子树，分割点后面元素一定在当前元素的右子树。然后加入下一个顶点，再把它当作分割点。如此递归的进行，直到二叉树建立完成。

---

# Translate:USACO/fence9

---

## Electric Fences 电网

Don Piele

译 by Charles.King

---

## 描述

---

在本题中，格点是指横纵坐标皆为整数的点。

为了圈养他的牛，农夫约翰建造了一个三角形的电网。他从原点  $(0,0)$  牵出一根通电的电线，连接格点  $[n,m]$  ( $0 \leq n < 32000, 0 < m < 32000$ )，再连接格点  $[p,0]$  ( $p > 0$ )，最后回到原点。

牛可以在不碰到电网的情况下被放到电网内部的每一个格点上（十分苗条的牛）。如果一个格点碰到了电网，牛绝对不可以被放到该格点之上。那么有多少头牛可以被放到农夫约翰的电网中去呢？

## 格式

---

**PROGRAM NAME:** fence9

**INPUT FORMAT:**

(file fence9.in)

输入文件只有一行，包含三个用空格隔开的整数：  $n, m$  和  $p$ 。

**OUTPUT FORMAT:**

(file fence9.out)

输出文件只有一行，包含一个整数，代表能被指定的电网包含的牛的数目。

## SAMPLE INPUT

---

7 5 10

## SAMPLE OUTPUT

20

### 分析:

给定顶点坐标均是整点（或正方形格点）的简单多边形，**皮克定理**说明了其面积  $A$  和内部格点数目  $i$ 、边上格点数目  $b$  的关系： $A = i + b/2 - 1$ 。

可以证明，一条直线 $((0,0),(n,m))$ 上的格点数等于  $n$  与  $m$  的最大公约数+1。即  $b=\gcd(n,m)+1$ 。 $\gcd(n,m)$ 为  $n$  与  $m$  的最大公约数。代入皮克公式，即可求出  $a$  的值。

方法二：局部枚举结果等于(线段 $(0, 0)--(N, M)$ 与线段 $(N, M)--(P, 0)$ )与  $X$  轴之间的格点数，我们分别计算两条线段与  $X$  轴之间的格点数，然后根据  $N$  与  $P$  的大小关系进行加减计算即可！在计算线段与  $X$  轴之间的格点数时，我们枚举每一个  $X$  轴上的坐标,得到对应于该  $X$  坐标下的所有格点数。

## Translate:USACO/rockers

### Raucous Rockers“破锣摇滚”乐队

译 by Maigo Akisame

### 描述

你刚刚继承了流行的“破锣摇滚”乐队录制的尚未发表的  $N(1 \leq N \leq 20)$  首歌的版权。你打算从中精选一些歌曲，发行  $M(1 \leq M \leq 20)$  张 CD。每一张 CD 最多可以容纳  $T(1 \leq T \leq 20)$  分钟的音乐，一首歌不能分装在两张 CD 中。

不巧你是一位古典音乐迷，不懂如何判定这些歌的艺术价值。于是你决定根据以下标准进行选择：

1. 歌曲必须按照创作的时间顺序在 CD 盘上出现。
2. 选中的歌曲数目尽可能地多。
3. 不仅同光盘上的歌曲写入时间要按顺序，前一张光盘上的歌曲不能比后一张歌曲写入时间要晚。

## 格式

---

**PROGRAM NAME:** rockers

**INPUT FORMAT:**

(file rockers.in)

第一行： 三个整数：N, T, M.

第二行： N 个整数，分别表示每首歌的长度，按创作时间顺序排列。

**OUTPUT FORMAT:**

(file rockers.out)

一个整数，表示可以装进 M 张 CD 盘的乐曲的最大数目。

## SAMPLE INPUT

---

```
4 5 2
4 3 4 2
```

## SAMPLE OUTPUT

---

```
3
```

分析：

Usaco 官方 DP：

This is a pretty straight-forward dynamic programming problem. The factors that determine whether we can put a song on a CD are:

- The length used up on the CD so far
- The length of the current song
- The last song that we put on the CD (because of the date restriction)





```

        if (dp[a][b][c] + 1 > dp[a][b + length[d]][d])
            dp[a][b + length[d]][d] = dp[a][b][c] + 1;
    }
    else {
        if (dp[a][b][c] + 1 > dp[a + 1][length[d]][d])
            dp[a + 1][length[d]][d] = dp[a][b][c] + 1;
    }
}
if (dp[a][b][c] > best)
    best = dp[a][b][c];
}
fprintf (out, "%d\n", best);
return 0;
}

```

### 枚举:

一道动态规划题，但观察数据规模，穷举就行了。穷举每首歌是否选取所有的组合可能( $2^{20}$ 种)，算出每种情况所有光盘上一共能存的歌曲数目，保留最大值即可。

对于穷举每首歌是否选取所有的组合可能，我采用了位运算的高效方法

limit=(1 << N)-1; for (i=0;i<=limit;i++)

然后 i 对应的每种状况计算能装进光盘中的最大的歌曲数目即可。

### 双重动态规划

---

用动态规划很容易的.

1、先利用三重循环计算出每个区间内的在背包重量内的最大值（利用 01 背包） 2、再利用三重循环，求解

F[I, J]代表前 I 个歌曲装在 J 个唱片的最优值，所以  

$$F[I, J] := \text{Max} \{F[K, J] + \text{Maxit}[K+1, I]\}$$

3、输出 F[N,M]