

A 蓝的无幂之夜

题目描述

作为八云蓝的式神，橙对计算机语言非常感兴趣，因此蓝教了她一些基础的C语言程序设计。

但是橙对函数的使用不是很熟练，还不能很好的理解函数的返回值是什么，因此她经常使用 `pow` 函数计算整型的幂，分不清 `abs` 和 `fabs` 的区别，导致代码出现了很多问题。

因此蓝留给橙一道作业，要她在不使用这几个函数的情况下计算整型的幂和绝对值：具体来说，给出整数 a, b ，计算 $|a^b - b^a|$ 的值

如果橙的程序中出现了 `math.h`，`pow`，`abs` 这三个字符串之一，蓝会狠心地打回橙的代码。

请你帮帮橙完成这份作业吧！

输入格式

不定组数据输入

每一行输入为两个正整数 a, b

输出格式

对于每一行输入，输出一行，一个正整数，为 $|a^b - b^a|$ 的值

样例

输入样例

```
1 1 3
2 2 2
3 5 6
```

输出样例

```
1 2
2 0
3 7849
```

数据范围

$0 < a, b \leq 100$

保证运算过程不会超过 `int` 的范围

输入不超过 100 行

Hint

八云紫悄悄给心爱的孙女递来一张小纸条：

不能使用 `pow` 和 `abs` 就自己写一个嘛，记得别用这两个名字，起点别的名字，例如

```
1 int myPow(int a, int b){
2     int ans=1;
3     //statement
4     return ans;
5 }
```

当然如果你实在想用的话也不是不可以，有很多绕过字符串匹配的方法，但是你有那功夫为什么不写一个 `myPow` 呢？

Author:星辰的微光

B 计算闪耀值

题目描述

“抓住它吧，你所期望的那颗星！”

九名怀着梦想的舞台少女，为了最终登上 *Top Star* 的宝座，在每场两人的舞台对决 *Revue* 中竞争着 *Position Zero*，向彼此展现自己的热情与闪耀。对每场 *Revue*，定义每位参与者的**闪耀值**为：

$\left\lfloor \frac{a! \bmod c}{b} \right\rfloor$ 的值，其中 a, b 分别是自己和对方的特征参数，而 c 是舞台环境参数， $a!$ 表示求 a 的阶乘， $\bmod c$ 表示对 c 取模，即计算除以 c 的余数， $\lfloor x \rfloor$ 表示对 x 向下取整。

现在有一场 *Revue* 即将上演于无法预测的命运之舞台，请你算出参演双方各自的闪耀值吧~。此外，当双方闪耀值之差的绝对值超过9时，请在下一行输出一个字符串 `Mabushi~`

输入格式

一行，3个正整数，以空格分隔，分别表示 a, b 与 c ，含义如题所述

输出格式

一行或两行

第一行两个整数，以空格隔开，第一个为 $\left\lfloor \frac{a! \bmod c}{b} \right\rfloor$ 的值，第二个为 $\left\lfloor \frac{b! \bmod c}{a} \right\rfloor$ 的值

若两数差的绝对值大于9，则在第二行输出 `Mabushi~`

输入样例

```
1 11 9 45989
```

输出样例

```
1 4926 3723
2 Mabushi~
```

数据范围

$1 \leq a, b \leq 10000, 1 \leq c \leq 46000$

Hint

模乘有如下性质： $(a \times b) \bmod c = ((a \bmod c) \times (b \bmod c)) \bmod c$

Author: 期待 *Revue* 的长颈鹿 *Asahi*

C Josephus函数

题目描述

Red 在思考排除第二个人的约瑟夫问题时抽象出了一个函数

$$J(kn + i) = \begin{cases} a_i & n = 0 \\ kJ(n) + b_i & n > 0 \end{cases} (i < k)$$

即对于 $N = nk + i$, $i = N \% k$ 有上式。

希望你能帮她计算这个函数的值。

输入格式

共 $T + 3$ 行。

第一行两个正整数 k, T 。

第二、三行每行 k 个整数，分别为 a_0, a_1, \dots, a_{k-1} 和 b_0, b_1, \dots, b_{k-1} 。

接下来 T 行，每行一个整数 n 。

输出格式

T 行，每行一个整数代表 $J(n)$ 的值。

样例输入

```
1 | 2 3
2 | 1 1
3 | -1 1
4 | 2
5 | 7
6 | 37
```

样例输出

```
1 | 1
2 | 7
3 | 11
```

数据范围

$1 < k \leq 100$, $0 \leq n \leq 10^9$ ，保证其余数值与运算中的结果均在 `int` 范围内。

Tips

样例的函数就是排除第二个人的约瑟夫问题的解哦

Aurthor : *Red*

D 地下墓穴的机关

题目描述

Olivaw在破解密码后，成功进入了最终的地下墓穴，但他眼前的台阶上有一系列机关。

他的朋友再次出现，指出需要在每一级台阶上放一定量的重物才能通过，规则为从第四级台阶开始时，每一级台阶上的重物数量是前三级之和。

此时，前三级台阶上分别被放置了一个重物，因此每级台阶上放置重物数 $M(n)$ 可以抽象为：

$$M(n) = \begin{cases} 1, & n = 1, 2, 3 \\ M(n-3) + M(n-2) + M(n-1), & n \geq 4 \end{cases}$$

现在，给你任意一级台阶的序号，请算出该台阶上需要放置重物的数量。**计算结果有可能很大，因此输出计算结果对1000000007取模的值。**

输入

一个正整数 n ，表示台阶序号。

输出

一个正整数 $M(n)$ ，表示该级台阶的重物数量。

输入样例

1 | 4

输出样例

1 | 3

数据范围

- $n \leq 100$

HINT

如果新学的“聪明”方法过不了，不妨试试以前的“笨”方法。

Author: Olivaw

E 毕业，但是忘记了总学分

题目描述

某一年，void即将从BUAA毕业。但是很不巧的是，他只记得自己的 GPA (绩点)而不记得自己的总学分了。确切的说，他忘记了自己某一门课的学分和成绩。

BUAA的 GPA 计算方式是这样的：对于一门课而言，如果这门课的得分在 60 分以下，那么这门课的 GPA 按照 0 计算。当然，能毕业的void肯定不会有哪门课挂掉的。

否则，这门课的 $GPA = 4 - 3 * (100 - x)^2 / 1600$ (x 为该门课的得分)。这是一个经过(60, 1.0)和(100, 4.0)的二次函数。而一名学生的 GPA ，是他**每门课**的 GPA 的**加权平均数**，权值为该门课的**学分**。

也就是：

$$\text{总绩点} = \frac{\sum_{\text{每门课}} (\text{该门课的绩点} * \text{该门课的学分})}{\sum_{\text{每门课}} \text{该门课的学分}}$$

在这里，我们认为，所有课的学分均在 0.5 和 6.0 之间，**只能为整数或者以.5结尾的小数**。学校对于学生毕业的总学分有一个要求，总学分**不得少于**这个最少值。void想知道，他忘掉的那门课究竟有多少种可能的学分，每种学分对应的这门课的成绩又是多少。

输入

输入共 n 行

第一行：void修的课的门数 n 、他的绩点 (GPA) m 、学校毕业的总学分要求 s

后续 $n - 1$ 行：每行包含两个数字，由一个空格隔开。第一个一位小数 w ，代表该门课的学分；第二个整数 x ，代表该门课的成绩

输出

输出共若干行

每一行输出两个数字，分别为忘掉的这门课的学分 w （保留一位小数）、对应的成绩 x （保留三位小数）

按照 w 递增的顺序输出

样例

样例1（不符合数据范围，仅仅作解释）

输入

```
1 5 3.77 25
2 6.0 94
3 5.0 87
4 4.5 89
5 6.0 90
```

输出

```
1 3.5 84.314
2 4.0 84.814
3 4.5 85.214
4 5.0 85.542
5 5.5 85.816
6 6.0 86.049
```

样例2

输入：

1	33	3.69	103.5
2	6.0	90	
3	2.0	87	
4	1.5	79	
5	2.0	90	
6	2.0	87	
7	1.5	79	
8	6.0	90	
9	2.0	87	
10	1.5	79	
11	5.5	90	
12	6.0	87	
13	1.5	79	
14	6.0	90	
15	3.0	87	
16	1.5	79	
17	2.0	90	
18	2.0	87	
19	1.5	79	
20	5.0	95	
21	6.0	90	
22	2.0	87	
23	1.5	79	
24	2.0	90	
25	2.0	87	
26	1.5	79	
27	6.0	90	
28	2.0	87	
29	1.5	79	
30	5.5	90	
31	6.0	87	
32	1.5	79	
33	4.5	90	

输出：

1	3.0	89.656
2	3.5	89.261
3	4.0	88.974
4	4.5	88.755
5	5.0	88.584
6	5.5	88.445
7	6.0	88.331

数据范围：

- $20 \leq n \leq 50$
- $1.00 \leq m \leq 4.00$
- $100.0 \leq s \leq 200.0$, 且 s 为整数或以 .5 结尾的小数
- $0.5 \leq s \leq 6.0$, 且 w 为整数或以 .5 结尾的小数

$60 \leq x \leq 100$, 且 x 为整数

且数据保证其他课的总学分小于要求的总学分 s

不需要考虑忘掉的这门课成绩小于 60 的情况

样例1解释：一共有6种情况满足题目要求

前面四门的总学分为21.5，那么最后一门的学分至少要是3.5，然后一直到6.0

六种情况下，这五门的总gpa均能满足为3.77

author:void

F 重炮的数学时间

题目描述

重炮今天也在好好学习！现在重炮遇到了一道难题：给出一个整数 n ，找出从 1 到 n 所有数字的所有排列方式。

因为昨晚熬夜熬得太晚，重炮现在要睡觉去了。你能写一道程序来帮助重炮解决这道问题吗？

输入

一个正整数 n 。

输出

输出从 1 到 n 所有数字的所有排列方式，按字典序排序。

输入样例

```
1 | 3
```

输出样例

```
1 | 1 2 3
2 | 1 3 2
3 | 2 1 3
4 | 2 3 1
5 | 3 1 2
6 | 3 2 1
```

数据范围

对于 100% 的数据， $3 \leq n \leq 9$

Author : *bluebean*

G 哪吒的水题（二）

题目描述

Simon 家和哪吒家住在笔直的 *Linda* 河的旁边，河位于 x 轴上，两人家的坐标分别为 $(x_1, y_1), (x_2, y_2), x_1 \leq x_2$ 。现在哪吒和 *Simon* 想在河的某处修建一座供水站，并修建**笔直**的水管通水到家里。由于 *Simon* 不是程序设计的助教，他使用的是低贱的 *Jason* 牌水管，而哪吒使用的是高贵的 *Kevin* 牌水管，**每单位距离价格**分别为 s_1, s_2 ($0 \leq s_1 \leq s_2$)。两人当助教的工资很少，他们想尽可能省点钱请 *Dora* 和 *Vanciry* 吃 $\text{Cra3\eta \text{Thues\eta}an}$ 。请聪明的你帮帮他们吧！

请确定供水站的位置，使得两人修建水管的总费用**尽可能少**，输出供水站的**横坐标**和这个**总费用**。

没有思路的同学建议仔细看Hint.

输入

共两行，第一行是四个整数，分别为 x_1, y_1, x_2, y_2 ，保证 $x_1 \leq x_2$

第二行是两个整数 s_1, s_2 ，保证不同时为 0。

输出

一行两个浮点数，**保留三位小数**，中间用一个空格隔开，分别表示修建供水站的横坐标和所需要的总费用

样例

输入 #1

```
1 | 0 1 1 1
2 | 1 1
```

输出 #1

```
1 | 0.500 2.236
```

输入 #2

```
1 | -3 1 2 -1
2 | 0 106
```

输出 #2

```
1 | 2.000 106.000
```

输入 #3

```
1 | 0 4 8 12
2 | 25 39
```


输出 #3

```
1 3.000 632.000
```

数据范围

$$|x_i|, |y_i| \leq 10000$$

$$0 \leq s_1 \leq s_2 \leq 10000$$

s_1, s_2 不同时为 0

Hint

总费用 s 是一个关于供水站横坐标的**先减后增函数**，可以选择使用**三分法**求该函数极值点；

也可以自己在纸上对 s 求导，然后用**二分法**求得导函数的零点。

代码模板

法一：三分法

l 和 r 为区间左右端点， s 为总费用关于供水站横坐标的函数，为一**先减后增**函数，用三分法求它的极小值点。部分代码如下：

```
1 mid1 = (2*l+r)/3;
2 mid2 = (l+2*r)/3;
3 while(r-l>eps) {
4     if(s(mid1) > s(mid2)) {
5         //此时供水站只可能在区间[mid1, r]上
6         //补全你的代码
7     }
8     else {
9         //此时供水站只可能在区间[l, mid2]上
10        //补全你的代码
11    }
12    mid1 = (2*mid1+mid2)/3;
13    mid2 = (mid1+2*mid2)/3;
14 }
```

此时 $l, mid1, mid2, r$ 之间差小于 eps ，一定精度范围下可视为相等，均为满足要求的供水站横坐标

法二：二分法

l 和 r 为区间左右端点， f 为 s 对供水站横坐标的导函数，为一**先负后正**函数，用二分法求函数 f 的零点。部分代码如下：

```
1 mid = (l+r)/2;
2 while(r-l>eps) {
3     if(f(mid) > 0) {
4         //补全你的代码
5     }
6     else {
7         //补全你的代码
8     }
9     mid = (l+r)/2;
10 }
```

此时 l, mid, r 之间差小于 eps ，一定精度范围下可视为相等，均为满足要求的供水站横坐标

PS: 感兴趣的同学可以思考一下本题和光的折射的联系

Author: 哪吒

H 前往苍盐海

题目描述

小羊学姐最近喜欢上了电视剧《苍兰诀》，尤其觉得里面的小兰花很可爱，想 *rua*。这一天，小羊学姐突然做梦发现自己穿越到《苍兰诀》里面了，她兴奋地想去找小兰花。

已知小羊学姐现在身处水云天（起点），小兰花在苍盐海（终点），且两地相距 n 公里。小羊学姐从今天开始计划前往苍盐海的旅途（但不一定今天就实际赶路），记今天为第一天，要求最早在第 m 天到达，最晚在第 l 天到达，且每天只能走**非负整数**公里。

现在小羊学姐想请你帮忙：对于给定的 n, m, l ，请输出所有满足以上要求的赶路方案，以及总方案数。

附：赶路方案输出顺序要求：

- 每一种方案输出 k 个非负整数，中间用空格隔开，表示每天的赶路公里数，且最终是在第 k 天到达苍盐海的（例如：0 1 2 表示在第三天到达的苍盐海，第一天走 0 公里，第二天走 1 公里，第三天走 2 公里）；
- 按照到达苍盐海时间从早到晚输出；
- 对于同一天到达苍盐海的赶路方案，优先输出前几天赶路公里数小的（例如：表示第三天到达苍盐海的，方案 0 1 2 要排在 0 2 1 前面）

输入

输入一行是三个正整数，分别代表 n, m, l

输出

若干行，表示所有满足以上要求的赶路方案，每行表示一共方案。
最后一行，一个数，表示总方案数。

输入样例

```
1 | 4 2 4
```

输出样例

1	0 4
2	1 3
3	2 2
4	3 1
5	0 0 4
6	0 1 3
7	0 2 2
8	0 3 1
9	1 0 3
10	1 1 2
11	1 2 1
12	2 0 2
13	2 1 1
14	3 0 1
15	0 0 0 4
16	0 0 1 3
17	0 0 2 2
18	0 0 3 1
19	0 1 0 3
20	0 1 1 2
21	0 1 2 1
22	0 2 0 2
23	0 2 1 1
24	0 3 0 1
25	1 0 0 3
26	1 0 1 2
27	1 0 2 1
28	1 1 0 2
29	1 1 1 1
30	1 2 0 1
31	2 0 0 2
32	2 0 1 1
33	2 1 0 1
34	3 0 0 1
35	34

数据范围

$1 \leq n \leq 20,$

$1 \leq m, l \leq 10$

保证本题所有数据都在 `int` 型范围内。

I 反约瑟夫

题目描述

1到n的n个整数按照序列 $[a_1, a_2, \dots, a_n]$ 围成一个环，从 a_1 开始，每隔m-1个数取出一个数加入另一个序列，然后从下一个数继续此操作，重复操作至所有数被取尽。最终恰好得到序列 $[1, 2, \dots, n]$ ，Uanu想问你开始时的序列 $[a_1, a_2, \dots, a_n]$ 是什么样的？

输入格式

1行输入。

第一行两个整数 n 和 m ，空格隔开。

输出格式

1行输出。

第一行输出 n 个整数，表示初始序列，空格隔开。

输入样例

```
1 | 5 2
```

输出样例

```
1 | 3 1 5 2 4
```

样例解释

每隔1个数取出一个，第一次取出1，然后从5开始操作，第二次取出2，以此类推，1,2,3,4,5依次被取出。

数据范围

$1 \leq n, m \leq 1000$

author: Uanu

J 二百由旬之一闪

无法斩断的东西，几乎不存在！——《东方妖妖梦》

题目描述

Konpaku Youmu 是白玉楼的园艺师，她正在打扫近两百由旬的 Saigyouji 家的庭院。

正值春季，樱花盛开，Youmu 准备用她削铁如泥的楼观剑来修剪樱花树。但是有些树实在是太硬了，连楼观都很难斩断，如果强行修剪，很可能会损坏刀刃。

庭院栽有 n 棵樱花树，修剪第 i 棵树需要的锋利度为 a_i ，楼观的初始锋利度为 S 。由于庭院太大，Youmu 不想来回往返，所以她会按遇到樱树的顺序来决定修不修剪这棵。

当 Youmu 遇到第 i 棵树：

- 如果决定修剪，并且楼观足够锋利 ($S \geq a_i$)，则可以轻松斩断；否则，强行修剪会导致刀变钝 (S 的值减少 1)；
- 如果不修剪，则无事发生。

当 Youmu 的楼观锋利度为 0 时，她会放下修剪的工作去给 Saigyouji Yuyuko 准备晚饭。如果能修剪更多的樱树，Yuyuko 大人一定会很高兴的！

输入

多组数据输入。

第一行一个整数 T ，表示数据组数。

对于每组数据：

第一行两个整数 n 和 S ，表示樱树的数量和楼观的初始锋利度；

第二行 n 个整数 $a_1, a_2 \cdots a_n$ ， a_i 表示 Youmu 遇到的第 i 棵樱树的修剪所需锋利度。

输出

每组数据一行一个字符串 s ，表示修剪樱树最多的一种方案。如果修剪第 i 棵，则 $s_i = 1$ ，否则 $s_i = 0$ 。

如果有多种可能，输出任意一种即可。

样例

输入

```
1 | 3
2 | 1 1
3 | 1
4 | 3 1
5 | 1 2 1
6 | 4 2
7 | 2 5 4 1
```

输出

```
1 | 1
2 | 101
3 | 1110
```

样例解释

对于第三组数据，Youmu 修剪了第 1, 2, 3 棵树后楼观的锋利度为 0，无法继续修剪。可以证明，这是修剪樱树最多的方案之一。

数据范围

$1 \leq T \leq 10^4, 1 \leq n \leq 10^5, 1 \leq S \leq 10^9, 1 \leq a_i \leq 10^9$ 。

对于每个测试点， $\sum n \leq 2 \times 10^5$ 。

K 岩王爷没有摩拉

题目描述：

岩王爷作为摩拉的生产者，原来是很有钱的，不过他现在失去了神之心，不能再生产摩拉了，只剩下 n 个钱包，每个钱包里有 a_i 摩拉，众所周知，岩王爷基本没有钱，所以他一般都需要两个钱包的钱加起来一起买单，现在岩王爷想知道 $\sum_{i=1}^n \sum_{j=i}^n |a_i + a_j - 1000|$

现在给出整数序列 a ，请你帮岩王爷求出这个式子的值。

输入描述：

- 第一行包括一个整数 $n (3 \leq n \leq 10^5)$ ，整数序列的长度。
- 第二行输入 n 个以空格分隔的整数 $a_i (0 \leq a_i \leq 1500)$

输出描述:

- 输出一个整数，表示该式子的值。

测试样例

输入

```
1 | 4
2 | 500 501 500 499
```

输出

```
1 | 8
```

HINT

- 暴力只能过一个点哦，要考虑时间复杂度。

author: HiDen