A 简单的计算器

题目描述

实现一个支持加减乘除四则运算的计算器

输入

第一行两个数字a，b，op，分别代表两个参与运算的数字，和执行运算的种类，下表给出了 op 和运算的对应关系

op	运算种类
0	+
1	-
2	*
3	/ ，整数除法，向下取整

输出

如果是加法运算，输出 `a + b = c`

如果是减法运算，输出 `a - b = c`

如果是乘法运算，输出 `a * b = c`

如果是除法运算，输出 `a / b = c`

其中 a, b, c 需要用具体数值替换

输入样例

```
2 3 0
```

输出样例

```
2 + 3 = 5
```

数据范围

$1 \leq a, b \leq 10000$

Hint

数据保证不会有除0的情况产生

B ljh想对齐

题目描述

老家伙*ljh*有一丢丢强迫症，现在他得到了6个整数，他想让他们尽可能美观的排列出来。

输入

一行，6个整数。

输出

依次输出这6个数，共两行，每行3个数，每个整数的宽度为6，不足**左边**补**空格**，每个数间以**一个空格**分开。

输入样例

```
123 1 2 3 4 345
```

输出样例

```
  123     1     2
    3     4   345
```

样例解释

```
***123 *****1 *****2
*****3 *****4 ***345
```

因为每个数的宽度为6，所以在$123, 345$之前补3个空格，在$1, 2, 3, 4$之前补5个空格，不要忘记每个数之间还有一个空格。

数据范围

任意整数$x < 10^7$。

HINT

这道题的难点在于如何实现输出特定宽度的数，其实`printf`函数里已经自带了这项功能，例如我们想要输出两个宽度为6的整数，中间用空格分开，就可以这样写：

```
printf("%6d %6d",a,b);
```

AUTHOR：ljh

C 山童的黑市

题目描述

Yamashiro Takane 在未经市场神的许可下私自组织并操纵了黑市！由于黑市经济活动的混乱，我们无法估计出商品的真实价值。Takane 的对头 Kawasiro Nitori 给你提供了一份商业机密，只要你能计算出其中商品的总价值，就能让市场恢复正常。

输入

第一行四个整数：商品种数 n，贵重商品价值 a_1，廉价商品价值 a_2，无用商品价值 a_3；均以一个空格分隔。

接下来 n 行，每行一个字符，`G` 代表贵重商品，`L` 代表廉价商品，其他非空白字符代表无用商品；一个整数 t，表示商品数量；以一个空格分隔。

输出

一个整数，表示所有商品的总价值。

样例

输入

```
5 85 50 0
L 26
9 59
< 43
L 40
G 22
```

输出

```
5170
```

数据范围

题目中所有整数均在 `int` 范围内，保证输出结果也在 `int` 范围内。

HINT

如何读入字符?

利用 `scanf` 函数读入四个整数后，如果直接读入字符，会读入第一行末尾的空白符，而不是下一行的字符。因此，可以像下面这样读入：

```
#include<stdio.h>

int main()
{
    char ch;
    //...
    scanf(" %c",&ch);//在%c前加一个空格，可以跳过任意数量的空白符直到读入一个非空白符
    //...
    return 0;
}
```

也可以扔掉行末的空白符：

```
#include<stdio.h>

int main()
{
    char ch,laji[20];
    //...
    gets(laji);//将行末空白符读入字符串中
    scanf("%c",&ch);
    //...
    return 0;
}
```

在任何时候都不建议使用 `getchar()` 来读入行末的空白符，因为行末的空白符数量不确定，在 OJ 上有可能是 `\n`，也有可能是 `\r\n`。

或者直接使用字符串读入：

```
#include<stdio.h>

int main()
{
    char s[20];
    //...
    scanf("%s",s);//此时s[0]为我们所要读入的字符
    //...
    return 0;
}
```

D 派蒙喜欢吃日落果

题目描述

派蒙很喜欢吃日落果，她现在有 m（$1 \leq m \leq 100$）个日落果，吃完一个日落果需要花费 t（$0 \leq t \leq 100$)分钟，吃完一个后立刻开始吃下一个。现在时间过去了s（$1 \leq s \leq 10000$)分钟，请问派蒙还有几个**完整**的日落果?

输入格式：

输入三个非负整数表示m, t, s。

输出格式：

输出一个整数表示答案。

输入样例：

```
50 10 200
```

输出样例：

```
30
```

Hint

如果你出现了 RE，不如检查一下被零除？

E 空白的黑白棋

题目描述

空正在进行一场赌上一切的黑白棋战斗。

在这场战斗中，每一枚空的棋子都代表了空“存在”的一部分。

现给出三个范围，说明编号在各个范围内的棋子分别代表空的什么“存在”（以一个数字表示），希望你能够根据需要查询的编号，告知对应的棋子代表的“存在”。

输入格式

输入共四行。

前三行格式均为 X_1, X_2, s，代表满足 $X\_1 \leq i \leq X\_2$ 的编号 i 的黑白棋代表的“存在”的数字是 s。

第四行为一个数，x，代表想要询问的黑白棋编号。

输出格式

输出共一行，一个数，为询问的黑白棋编号对应的“存在”对应的数字。

如果询问的编号不在范围内，输出None四个字母。

样例数据

数据点#1

输入

```
1 2 1
3 7 2
8 13 3
11
```

输出

```
3
```

数据点#2

输入

```
2 3 1
4 7 2
8 13 3
1
```

输出

```
None
```

数据规模

保证总成立
$1 \leq X\_1 \leq X\_2 \leq 13$
$1 \leq x \leq 13$
$1 \leq s \leq 9$

保证前三行六个代表范围的整数 X_1 与 X_2 严格单调递增，即第一行的 X_2 严格小于第二行的 X_1 ,第二行的 X_2 严格小于第三行的 X_1 。

Author: Untitled

F O！平均分鸽

题目描述

Op 喜欢鸽子和笼子，他更喜欢平均分配。现在他有许多兑换券，上面写有一个 0，1，2…… （`int` 范围内）的非负整数，数值的含义是可以用此兑换券换取等数额的鸽子或笼子。现在 Op 的想法是用数值最大的**两**张兑换券兑换鸽子，数值最小的**一**张兑换券兑换笼子。

现在请你帮他计算：**按平均分鸽的方式安排鸽子和笼子，至少会有多少只鸽子出现在同一个笼子中**?

每张兑换券只能使用一次

输入

一组表示 Op 拥有的兑换券数值的数据，包括多行数据输入，以 `-1` 表示结束（数据不包括 `-1` ）

输出

对于每组数据，输出一行，表示结果（至少会有多少只鸽子出现在同一个笼子中）

Op 可以暂时没有鸽子，但 Op 不能**没有笼子**来分鸽子，正如西方不能没有耶路撒冷。没有笼子他将不能完成装入操作。

如果无法完成将鸽子装入笼子中这一操作，请输出 `oh O!`

输入样例

```
3
2
10
-1
```

输出样例

```
6
```

样例解释：

按照 Op 的想法：他会兑换 $10+3$ 只鸽子， 兑换 2 个笼子，至少有一个笼子中有 6 只鸽子

数据范围

输入的数据和计算结果确保在非负整数范围内，且不会超过 `int`

输入的数据个数大于等于 `3`

结束标志数字为 `-1`

G 字符处理机器

题目描述

Uanu最近设计一个字符处理机器，机器读入一个字符 `c` 并进行如下处理：

- `c` 如果是英文字母，则将其大小写转换后输出（即大写转小写，小写转大写）。
- `c` 如果是数字 `n`，无输出且下一个英文字母处理后重复输出 n 次（无空格）。（以最近一次输入的数字为准）
- `c` 如果既不是英文字母也不是数字，则输出 `?*&\!_//\a@\\\r\n!//\\"_"/\\^!~zZ`。

输入格式

共一行，一个整数 n 和连续 n 个字符，表示机器读入的 n 个字符，整数和字符间空格隔开。

输出格式

多行，每行根据情况输出：

- 一个转换后的英文字母；
- 多个重复的转换后的英文字母；
- `?*&\!_//\a@\\\r\n!//\\"_"/\\^!~zZ`；

读入数字时无需换行。

输入样例

```
8 1a2G(c)5
```

输出样例

```
A
gg
? *&\! _/\a@\\\r\n! /\\"_"/\\^! ~zZ
C
? *&\! _/\a@\\\r\n! /\\"_"/\\^! ~zZ
```

HINT

在输出如 `\`、`"`、`'` 字符时，需要在该字符前增加一个反斜杠 `\`。
例如要输出 `\`：

- `printf("\");`（❌）
- `printf("\\");`（✅）

字符中无空格。

author: Uanu

H 用代码来写歌！

题目描述

Jerydeak 爱好特别广泛，除了做游戏、写代码，他还喜欢玩音乐。这天，他有一个想法：通过写代码来制作音乐。

说干就干，他想**通过输入音符，生成能产生音乐的代码。**

他要输入 n 个音符，每个音符由频率 f 和时值 t 两个因素构成，他想通过输入每个音符的音高 p 和长度 l 来**生成一段可以播放音乐的代码。**

具体来说，音高 p 与频率 f 之间对应关系规定如下：

音高 p 值	频率 f 值
0	0
1	523
2	587
3	659
4	698
5	784
6	880
7	988

时值 t 和长度 l 由速度基数 s 决定，满足关系：$t = l * s$；

输入

第一行，两个整数 n，s，分别表示音符个数和速度基数。

接下来 n 行，每行两个整数 p，l，分别表示一个音符的音高和长度。

输出

你需要生成这样一段代码：

```
#include<stdio.h>
#include<stdlib.h>

int main(){
    // music
}
```

`// music` 的注释处为多行，每一行表示一个音符，其由四个空格与 `_beep(f, t);` 构成，其中 `f` 表示该音符的频率，`t` 表示该音符的时值（注意：`t` 前有一个空格）。

样例

输入 #1

```
14 500
1 1
1 1
5 1
5 1
6 1
6 1
5 2
4 1
4 1
3 1
3 1
2 1
2 1
1 2
```

输出 #1

```
#include<stdio.h>
#include<stdlib.h>

int main(){
    _beep(523, 500);
    _beep(523, 500);
    _beep(784, 500);
    _beep(784, 500);
    _beep(880, 500);
    _beep(880, 500);
    _beep(784, 1000);
    _beep(698, 500);
    _beep(698, 500);
```

```
    _beep(659, 500);
    _beep(659, 500);
    _beep(587, 500);
    _beep(587, 500);
    _beep(523, 1000);
}
```

数据范围

$1 \leq n \leq 100$，$300 \leq s \leq 500$，对每个 p，有 $0 \leq p \leq 7$；对每个 l，有 $1 \leq l \leq 16$。

HINT

输出生成的代码能运行噢！

Author: Jerydeak

I 组三角形

题目描述

你和派蒙来到了梦中的桓那兰纳，在这里你们遇到了兰罗摩。兰罗摩给派蒙介绍了一个游戏：

兰罗摩拿出长度为1到 n 的 n 条木棒，派蒙和他轮流从中取出一根，直到只剩下3根木棒。若剩下的三根木棒可以拼成一个三角形（即三根木棒的长度 a, b, c 满足 $|a-b| < c < a+b$），那么先手的派蒙获胜，否则兰罗摩获胜。

兰罗摩告诉派蒙，如果派蒙获胜就会把自己珍藏的"暴葬"送给她。派蒙很想要"暴葬"，但是派蒙和超级白化漂浮灵一样笨，你能告诉他他是否能获胜吗？因为兰罗摩很珍惜他的"暴葬"，所以他会采用最优解。

输入格式

第一行一个整数 T，代表游戏进行的次数。

第 2 到 $T+1$ 行，每行一个整数 n ,代表该次游戏木棒的数量。

输出格式

输出 T 行。

对于第 i 行，若派蒙能在第 i 次游戏获胜，则输出 `Yes`，否则输出 `No`。

样例输入

```
1
7
```

样例输出

```
No
```

数据范围

对于100%的数据，$0 < T < 10$，$3 < n < 10^3$

Aurthor : Red

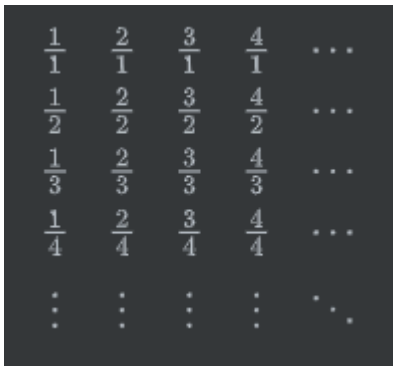
J 有理数

题目背景

在离散数学（2）中，大家将会学到有关无穷集的大小的比较的问题。可以证明，有理数集和整数集是等势的。为证明有理数集和整数集等势，我们只需要证明正有理数集和正整数集等势。这道题与其中的一个证明方法有关。

题目描述

为此，我们考虑给所有的正有理数进行一个“编号”。考虑构造以下的无穷方阵（显然这个方阵涵盖所有正有理数）：



然后，我们按照从右上到左下的对角线依次为这些有理数“编号”，依次是：

第一个对角线	第二个对角线		第三个对角线			第四个对角线				第五个对角线					...
1/1	2/1	1/2	3/1	2/2	1/3	4/1	3/2	2/3	1/4	5/1	4/2	3/3	2/4	1/5	...

$$\frac{1}{1},\frac{2}{1},\frac{1}{2},\frac{3}{1},\frac{2}{2},\frac{1}{3},\frac{4}{1},\frac{3}{2},\frac{2}{3},\frac{1}{4},\frac{5}{1},\frac{4}{2},\cdots$$

但是，其中一些分数是重复的（如$\frac{2}{2}$和$\frac{1}{1}$），因此我们要删除所有的形如$\frac{a}{b}$但$\gcd(a,b)\neq 1$的分数。最终分数与“编号”对应关系如下：

编号	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**...**
分数	$\frac{1}{1}$	$\frac{2}{1}$	$\frac{1}{2}$	$\frac{3}{1}$	$\frac{1}{3}$	$\frac{4}{1}$	$\frac{3}{2}$	$\frac{2}{3}$	$\frac{1}{4}$	$\frac{5}{1}$	$\frac{1}{5}$	...

本题的目标是，给出a和b的值，求出正分数$\frac{a}{b}$的编号（数据保证$\gcd(a,b)=1$）。

输入输出说明

输入说明

一行，两个正整数a和b，保证$\gcd(a,b)=1$。

输出说明

一行，一个正整数x，表示$\frac{a}{b}$的编号是x。

输入输出样例

样例输入1

```
77 23
```

样例输出1

```
3013
```

样例输入2

```
2713 287
```

样例输出2

```
2735464
```

样例输入3

```
8671 6329
```

样例输出3

```
68392003
```

数据范围与提示

gcd函数：表示最大公约数。ppt例1-9的代码可用，但是建议使用更快速的方法计算。

对于40%的数据，保证$a+b\leq 200$。这部分数据可以直接参考题意写双重循环+gcd验证，且gcd验证可以直接使用ppt例1-9的代码，直到找到题述分数为止，然后直接输出遍历到的gcd=1的有效分数的个数，即为编号。

对于80%的数据，保证$a+b\leq 4\times 10^3$。这部分数据也可以直接参考题意写双重循环+gcd验证，但是需要考虑更快速的gcd代码。

对于100%的数据，保证$a+b\leq 10^6$。这部分数据超出了第一次课的要求，仅留给有数论或信竞基础的同学思考。其他同学也可以在学习更多知识后再来思考本题。

参考代码

采用双重循环+gcd验证的同学可以参考以下代码（仅为部分代码）

```
int a, b, x = 0, flag = 0;
scanf("%d%d", &a, &b);
for(int s = 2; ; s++) // s 表示 i+j 的和
{
    for(int j = 1; j < s; j++)
```

```
    {
        int i = s - j; // 遍历到分数 i/j
        int g; // g = gcd(i,j)

        // do something

        if(g == 1) x++; // 说明这是一个符合要求的分数
        if(i == a && j == b) // 找到了!
        {
            flag = 1; break; // flag置1，退出循环
        }
    }
    if(flag) break;
}
printf("%d\n", x);
```

Author: Toby