N Mogg的简单数论

时间限制：1000ms 内存限制：65536kb

通过率：18/193 (9.33%) 正确率：18/541 (3.33%)

题目描述

给定两个数`a b`, 求正整数`x y`使得 $x + y = a$, $lcm(x, y) = b$

lcm是最小公倍数

输入格式

多组输入数据

每组数据一行两个数字 `a b` $(1 \le a \le 2000, 1 \le b \le 10^6)$

输出格式

每组数据输出一行，如果存在xy则输出`x y`(由小到大)，否则输出`No Solution`

输入样例

6 8 798 10780

输出样例

No Solution 308 490

Hint

$lcm(x, y) = x \times y \div gcd(x, y)$

gcd最大公约数

解题思路

设 $g = gcd(x, y)$

$$\begin{cases} x + y = a \\ lcm(x, y) = b \end{cases} \Rightarrow \begin{cases} k_1 \times g + k_2 \times g = a \\ k_1 \times k_2 \times g = b \end{cases} \Rightarrow \begin{cases} k_1 + k_2 = \frac{a}{g} \\ k_1 \times k_2 = \frac{b}{g} \end{cases}$$

其中k_1 k_2为正整数且互质

因为k_1 k_2互质，不难推出$k_1 + k_2, k_1 \times k_2$互质（反证法总能证吧）

所以$\frac{a}{g}, \frac{b}{g}$互质

所以$gcd(a, b) = g = gcd(x, y)$

解方程组

$$\begin{cases} k_1 + k_2 = \frac{a}{g} \\ k_1 \times k_2 = \frac{b}{g} \end{cases}$$

即可。（只需要整数解）

可以转换成一个二元一次方程求根。

反证法证明$k_1 k_2$互质时，$k_1 + k_2, k_1 \times k_2$互质：

假设$k_1 + k_2, k_1 \times k_2$不互质，则$k_1 + k_2, k_1 \times k_2$必然存在公约数（不是最大公约数）$d$，且$d$为$k_1$的约数或者$d$为$k_1$的约数

若d为k_1的约数，则$\frac{k1+k2}{d} = \frac{k1}{d} + \frac{k2}{d}$

因为k_1, k_2互质，所以k_2与d必然不能整除，则$\frac{k1}{d} + \frac{k2}{d}$为小数，这与$d$是$k_1 + k_2, k_1 \times k_2$公约数矛盾。

同理可以证明d为k_2的约数时也不成立。

所以$k_1 + k_2, k_1 \times k_2$互质

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

int gcd(int a, int b)
{
    return b == 0 ? a : gcd(b, a%b);
}

int main()
{
    int a, b;
    while (~scanf("%d%d", &a, &b))
    {
        int g = gcd(a, b);
        a /= g;
        b /= g;
        int dit = a * a - 4 * b;
        if (dit < 0)
        {
            printf("No Solution\n");
            continue;
        }
        int t = sqrt(dit);
        int k1 = a + t, k2 = a - t;
        if (t*t != dit || (k1 & 1) != 0 || (k2 & 1) != 0)
        {
            printf("No Solution\n");
        }
        else
        {
            printf("%d %d\n", k2 / 2 * g, k1 / 2 * g);
```

```
        }
    }
    return 0;
}
```