

F 盗墓笔记之怒海潜沙

题目描述

西沙水下，一阵暗流，Thor 与天真一行人被冲散了。

幸运的是，他被冲到了一扇门前。

不幸的是，刚刚到达这扇门前，门就合上了。上面的机关被重置为初始状态。

经过仔细的观察，Thor 发现，机关为一个圆盘，分为内外两圈。内圈均匀刻着 a 个小格，外圈均匀刻着 b 个小格，每个刻度有一个独一无二的图腾与之对应。

他还发现，每经过一分钟，内外两圈会同时顺时针旋转一个单位。从身边墙壁上的铭文，他得知，当轮盘再次回到初始状态时，门会重新打开。他想知道从门被合上到再次打开的这一个轮回需要多少时间，这个任务就交给你了。

输入

多组测试数据。

对于每组测试数据，输入两个数 a, b ($1 \leq a, b \leq 30000$)，用空格隔开，具体含义见题目描述。

输出

对于每组数据，输出一行，包含一个整数，表示一个轮回需要的时间。

输入样例

3 4

4 8

12 16

输出样例

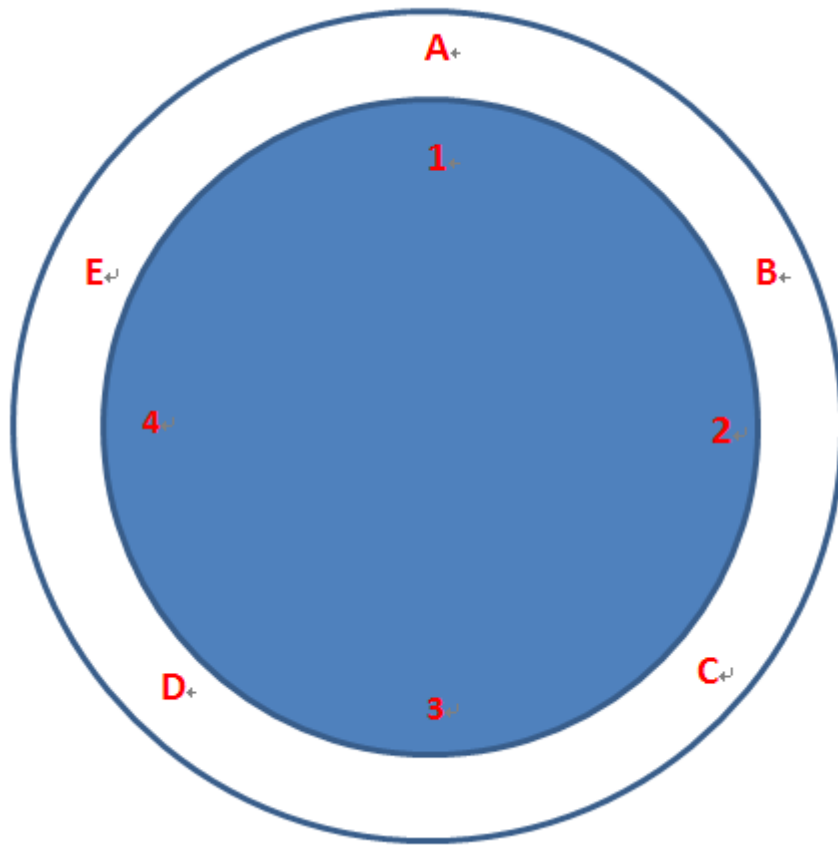
12

8

48

HINT

机关？大概.....就是这个样子吧.....OTL



解题分析：

本题先要理解题目描述的问题到底是什么。内外同时同向转动，要让二者又回到初始位置，慢圈和快圈要在相同时间内都转过几个整圈，就能想到，最短时间就是二者周期的最小公倍数。这里要留心看题干，最终状态并不只是二者的相对位置重合，而是二者最终都要回到初始形态，即上图中 A 和 1 都在十二点钟位置的状态。(你想啊，这一个大门，肯定是绝对位置回到起点嘛~)

知道了要求最小公倍数，就容易解答了。最小公倍数等于两数乘积再除以两数的最大公约数，而最大公约数可以用辗转相除法解决。建议自定义一个求最大公约数的函数，使主函数比较简洁。

参考代码：

```
#include <iostream>
using namespace std;
int zdgy(int a, int b)    //求最大公约数的函数
{
    int c;
    while(b%a!=0)
    {
        c=b%a;
        b=a;a=c;
    }
    return a;
}
int main()
{
    int a,b;
    while(cin>>a>>b)
    {
        cout<<a*b/zdgy(a,b)<<endl;
    }
    return 0;
}
```

P.S.最大公约数——辗转相除的递归解决方法：

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

鉴于同学们刚刚学过递归，可以考虑一下上面这个辗转相除是如何实现的。