# 第五次上机解题报告

----14211062 杨晨

## 题目 A：jhljx 学 gcd

Problem Description

大家都知道 gcd 是最大公约数的意思。jhljx 准备开始学习 gcd 了。他想要求出 n 个数的最大公约数 gcd 和最小公倍数 lcm。请你帮帮他。

Input

输入多组数据。
每组数据两行，第一行为一个正整数 n,表示有多少个数(2<=n<=20)。
第二行有 n 个正整数，每个数之间用空格隔开。

Output

输出这 n 个数共同的最大公约数和最小公倍数（保证结果在 int 范围内）。

Sample Input

**2**
**9 15**
**3**
**24 60 18**

Sample Output

**3 45**
**6 360**

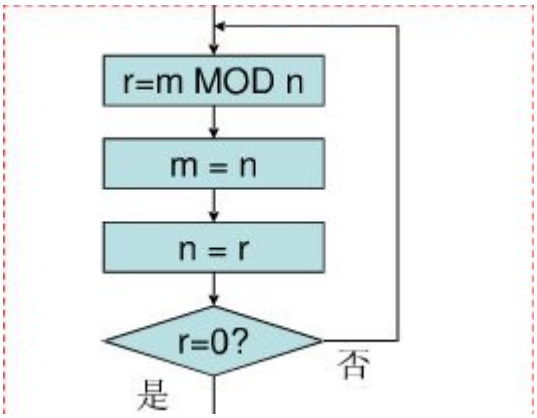
解析：求 n 个数的最大公约数：可先求出 n 个数中最小的数，然后用这 n 个数去除以从 1 到最小数中的每个数，能够被所有数整除的最大数就是最大公约数；

求 n 个数的最小公倍数：先求两个数的最小公倍数，再求所得的数与第三个数的最小公倍数，如此顺次求下去，便可求得 n 个数的最

小公倍数。

求两个数的最小公倍数：最小公倍数=（x*y）/最大公约数；

求两个数的最大公约数：辗转相除法




代码：

```
#include<iostream>
#include<cstdio>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    int n;
    while(cin>>n)
    {
        int a[n];
        for(int i=0;i<n;i++)
        cin>>a[i];
        int min;
        min=a[0];
        for(int i=1;i<n;i++)
        {
            if(a[i]<min)
            min = a[i];
        }//求 n 个数中最小的数
        int gcd=1;
        for(int counter=2;counter<=min;counter++)
        {
            int num=0;
            for(int i=0;i<n;i++)
```

```
            {if(a[i]%counter==0)
            num++;}//验证是否每个数都能整除 counter
            if (num==n)
            gcd=counter;
        }//求 n 个数的最大公约数
        int lcm(int x,int y);
        int gbs;
        gbs=lcm(a[0],a[1]);
        for(int i=1;i<(n-1);i++)
        gbs=lcm(gbs,a[i+1]);//求前面两个数的最小公倍数与第三个数
的最小公倍数
        cout << gcd << " " << gbs << endl;
    }
}
int lcm(int x,int y)//求两数的最小公倍数
{
    int max,min,lcm;
    if(x>y)
    {max = x;min = y;}
    else
    {max = y;min = x;}
    int r;
    while((r=(max%min))!=0)
    {
        max=min; min = r;
    }//求两数的最大公约数
    lcm = x * y / min;
    return lcm;//求两数的最小公倍数
}
```

题目 **B**： jhljx 学素数

# Problem Description

函数是一个重要的知识点。**jhljx** 一改丧心病狂的风格，来点小清新。
他让你用函数实现判断一个数是否为素数。

# Input

输入多组数据。
输入一个非负整数 n。(保证 n 在 long long 范围内,但不会很大)

# Output

如果这个数是素数，输出"jhljx is good!"，否则输出"jhljx is sangxinbingkuang!"。

# Sample Input

1
2

# Sample Output

jhljx is sangxinbingkuang!
jhljx is good!

解析：

题目很简单，就是一个判断素数的问题。依次验证从 2 到根号 n 的每个整数是否能被该数整除，若都不能，则该数是素数，否则不是。

代码：

```
#include<iostream>
#include<cstdio>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    void sushu(long long);
    long long n;
    while(cin >>n)
    sushu(n);
}

void sushu(long long n)//判断一个数是不是素数

{
```

```
    if(n==1)
    cout<<"jhljx is sangxinbingkuang!"<<endl;//1 必然不是素数
    else
    {
        int counter=0;
        for(int i=2;i<=(sqrt(n));i++)
        {
            if(n%i==0)
            counter++;
        }//判断从 2 到根号 n 各整数是否能被 n 整除
        if(counter==0)
        cout<<"jhljx is good!"<<endl;
        else
        cout <<"jhljx is sangxinbingkuang!"<<endl;
    }
}
```

题目 C：jhljx 学下棋

# Problem Description

**jhljx** 最近喜欢上了下棋，他要和 **Last_Day** 下棋。
**Last_Day** 给了他一个 **n*n** 的棋盘。**jhljx** 决定在棋盘上放上小兵。如果小兵放在（**x,y**）位置，那么他会攻击处在（**x-1,y**）,**(x+1,y)**,（**x,y-1**）,**(x,y+1)**在四个位置的棋子（如果这几个位置存在）。请问 **jhljx** 最多可以放多少个小兵，保证他们不会相互攻击。

# Input

输入多组数据。
每组数据一行，为一个数 **n**。**(**保证 **n** 在 **int** 范围内**)**

# Output

输出最多可放置的小兵的个数。

# Sample Input

1
2
3

# Sample Output

1
2
5

解析：

多画几次，发现这道题有一定规律。

当 n 为奇数时，结果=1+3+...+n+...3+1，即为(n^2+1)/2;

当 n 为偶数时，结果=1+3+...+(n-1)+(n-1)+...+3+1,即为(n^2)/2.

代码：

```
#include<iostream>
using namespace std;
int main()
{
    long long n,max;
    while(cin>>n)
    {
        int r;r=n%2;//判断 n 的奇偶；
        if(r==0)
            max=(n*n)/2;//n 为偶；
        else
            max=(n*n+1)/2;//n 为奇；
```

```
            cout << max << endl;
        }
}
```

题目 **D**：jhljx 的强迫症

Problem Description

**jhljx** 最近有点不太正常，他觉得自己貌似患上了一种奇奇怪怪的病，这种病好像叫做强迫症。。噗。。一天，树荫姐给了 **jhljx** 两个数 **n** 和 **m**,树荫姐说我们来做 **n** 和 **m** 的模运算吧。。

**jhljx** 叫道：“好吖好吖”。**jhljx** 虽然数数数不清，但他不喜欢别人 **chaofeng** 他数数数不清。。 

于是，**jhljx** 决定证明给你们看。**jhljx** 拿着 **n** 这个数左右把玩，他不断地对 **n** 累加，于是得到了 **n,2n,3n,4n**……拿着许许多多的数 **jhljx** 很开心。但是他想知道这些数模上 **m** 的结果

**(**举个例子吖，就是 **n%m,2n%m,3n%m,**……**)**是不是能够得到 **0~m-1** 之间的所有数,只有得到了 **0~m-1** 之间的所有数 **jhljx** 才心满意足，如果没有得到，他连觉都睡不好。

Input

输入多组测试数据直到文件结束。

每组测试数据只有一行，为 n 和 m 的值。n 和 m(n>0,m>0 且保证 n 和 m 在 int 范围内)之间用空格隔开。

Output

如果 n 的所有倍数模上 m 的值能够取遍 0~m-1 之间的所有数，输出“jhljxshidadoubi”,反之，输出“shuishuowoshidadoubi”。

Sample Input

```
3 5
```

Sample Output

```
jhljxshidadoubi
```

Hint

童鞋快看这里。

3+0=3,3%5=3;

3+3=6,6%5=1;

3+3+3=9,9%5=4;

3+3+3+3=12,12%5=2;

3+3+3+3+3=15,15%5=0;

3+3+3+3+3+3=18,18%5=3;

m=5,这些余数取到了 0，1，2，3，4，满足！get!

解析：

这道题确实不好想。看到它的第一反应一般会是去开数组，可是最后发现虽然能出结果但是交上去后却 Runtime Error。于是就换了一种思路。可以看出，如果 n 和 m 有最大公约数 r 的话，那么 n 的倍数与 m 求模，所得结果也只能是 r 的倍数，而不会出现其他情况，所以若想遍历 0~m-1 所有数，只需让两个数最大公约数为 1 即可，即两个数互质。求两数最大公约数的方法上边已讲过了。

代码：

```
#include<iostream>
using namespace std;
int main()
{
    int n,m;
    while(cin>>n>>m)
    {
        int zuida,zuixiao;
        if(n>m)
        {zuida=n;zuixiao=m;}
        else
        {zuida=m;zuixiao=n;}//两数排序；
```

```
            int r;
            while((r=zuida%zuixiao)!=0)
            {
                zuida=zuixiao;
                zuixiao=r;
            }//求两数最大公约数；
            if(zuixiao==1)
                cout<<"jhljxshidadoubi"<< endl;
            else
                cout << "shuishuowoshidadoubi"<< endl;
        }
}
```

题目 E：汉诺塔再度来袭

Problem Description

汉诺塔（又称河内塔）问题是源于印度一个古老传说的益智玩具。大梵天创造世界的时候做了三根金刚石柱子，在一根柱子上从下往上按照大小顺序摞着 64 片黄金圆盘。大梵天命令婆罗门把圆盘从下面开始按大小顺序重新摆放在另一根柱子上。并且规定，在小圆盘上不能放大圆盘，在三根柱子之间一次只能移动一个圆盘。


假设三根柱子分别是 A,B,C。盘子编号为 1，2，3……n,最开始时，按照编号从小到大的顺序放在 A 柱子上。n 号盘子在最下方，1 号盘子在最上方。

Input

输入多组数据。

每组数据一个 n,表示黄金圆盘的个数。(1<=n<=20)

Ouput

输出需要移动的步数和移动的具体方案。详细请参见样例。

比如 **1 A->C** 表示将 **1** 号盘子从 **A** 柱子上移到 **C** 柱子上。

Sample Input

**1**

**2**

Sample Output

**1**

**1 A->C**

**3**

**1 A->B**

**2 A->C**

**1 B->C**

Hint

样例解释：

**Samle 1**：输出的结果 **1 A->C** 数字和字母之间有一个空格

本题请用 **scanf** 和 **printf** 进行输出，用 **cin** 和 **cout** 会超时。

解析：

这道题和练习赛中的几乎相同，只是多输出了一个步数而已。设盘数

为 n，步数=2^n - 1。具体移动过程只要调用递归函数就可以实现。

递归函数中既调用了自己，又调用了 move 函数，hanoi 函数将权利一级一级下放，直至到最后一级然后调用 move 函数。

代码：

```
#include<iostream>
#include<cstdio>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    void hanoi(int,char,char,char);
    int n;
    while(~scanf("%d",&n))
    {int pow = 1;
    for(int i=1;i<=n;i++)
    pow = pow * 2;

    printf("%d\n",pow-1);//计算移动步数

    hanoi(n,'A','B','C');//移动方法}

}

void hanoi(int n,char one,char two,char three)//具体移动方法

{
    void move(char,char);
    if(n==1)
    {
        printf("1 ");
        move(one,three);
    }
    else
    {
        hanoi(n-1,one,three,two);

        printf("%d ",n);//移动的盘子号码

        move(one,three);
        hanoi(n-1,two,one,three);

    }//运用递归，层层下放
```

```
}
void move(char x,char y)//移动函数
{
    printf("%c->%c\n",x,y);
}
```

题目 F：jhljx 学斐波那契数列(I)

Problem Description

**jhljx** 听说你们学了斐波那契数列，于是他想考考你们。他想问你斐波那契数列的第 **n** 项是多少。

有人不知道斐波那契数列？

它就是 **1，1，2，3，5，8.....**这样的一个数列。

Input

输入多组数据。

每组数据输入一个 **n**。

Ouput

输出斐波那契数列中的第 **n** 个数。

Sample Input

**1**

**2**

**3**

Sample Output

1

1

2

解析：

这道题很简单，直接用递归一级一级求下去就可得最终结果。

代码：

```
#include<iostream>
#include<cstdio>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    unsigned long feibonaqi(unsigned long);
    unsigned long n;
    while(cin>>n)
    cout<< feibonaqi(n) << endl;
}

unsigned long feibonaqi(unsigned long n)//求斐波那契数列第 n
项
{
    if(n==1||n==2)
    return 1;
    else

    return feibonaqi(n-1)+feibonaqi(n-2);//递归

}
```

题目 G：

# Problem Description

数列是一个神奇的存在。
现在定义一个数列:f(n)=|f(n-1)-f(n-2)|
给你 f(1)和 f(2)
//求 f(n)
年轻人好好刷题别做梦了，怎么会那么简单。
KamuiKirito 告诉你这个数列会出现的数字个数为有限个。
求该数列会出现的数字个数。

# Input

输入多组数据。
每组数据为两个整数 a,b，代表 f(1)和 f(2)。(0<=a,b<=10^18)

# Output

每组数据输出一行，为会出现的数字个数。

# Sample Input

2 1
4 6

# Sample Output

3
4

解析：

这道题挺难想的，我做的时候也是偶然间发现了一个规律。本来看到 f(x)=|f(x-1))-f(x-2)|的时候想用更相减损做，但是交上去之后超时了，于是我便开始尝试用另一种求最大公约数的方法，即辗转相除法。把辗转相除所得各个商加起来实际上就是更相减损的步数。

例如：153 和 123：

利用更相减损则有：153-123=30；123-30=93；93-30=63；63-30=33；33-30=3；30-3=27；27-3=24；24-3=21；21-3=18；18-3=15；15-3=12；12-3=9；9-3=6；6-3=3；3-3=0.共计 15 次；

而利用辗转相除法则有：　153/123=1...30；　123/30=4...3；30/3=10...0.共计 1+4+10=15 次.

利用两种方法可以得到相同的结果，而用辗转相除则可以大大地减少循环次数，节省时间。

此外，本题还需注意几点。比如，特殊情况的讨论。如果 a=b=0 或者是 a,b 其中一个为 0 的话应该单独讨论，否则 0 在除数的位置上会造成 Runtime Error。还需要注意题干中的变量范围，由于 a,b<=10^18,应声明为 long long 类型，否则会造成 Wrong Answer.

代码：

```
#include<iostream>
using namespace std;
int main()
{
    long long a,b;
    while(cin>>a>>b)
    {
        if(a==0&&b==0)
            cout<<"1"<<endl;
```

```
        else if(a==0&&b!=0)
            cout<<"2"<<endl;
        else if(a!=0&&b==0)
            cout<<"2"<<endl;//以上为特殊情况的讨论；
        else
        {
        long long da,xiao;
        if(a>b)
        {da=a;xiao=b;}
        else
        {da=b;xiao=a;}//a,b 按大小顺序排出；
        long long r;long long s=1+da/xiao;
        while((r=da%xiao)!=0)
        {
            da=xiao; xiao=r;
            s=s+da/xiao;
        }//辗转相除法；
        cout << s << endl;
        }
    }
}
```