

2014 级 C++第五次上机解题报告

14211076 马国瑞

首先是对以下内容的说明。

所有题目我都没有使用数组，有一两题做法个人认为比较巧妙，谨此给大家分享。

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 ($2 \leq n \leq 20$)。

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
```

二、解题思路

如果是求 2 个数的 gcd 和 lcm 那么 so easy，问题是，这是求 n 个数的，难度上来了些。

鉴于本人在此次上机没有使用数组，我的思路是这样的：

1. 先求头两个数的 gcd 和 lcm；
2. 接着求前 2 个数 lcm 与第 3 个数这两个数的 gcd 与 lcm；

3. 依此类推，求前 **n-1 个数 lcm** 与第 **n 个数** 这两个数的 gcd 与 lcm；
4. 将前面所求的所有 gcd 取最小值（取最小值的方式是每次计算，如果新 gcd 比老的小就替代之，注意求 lcm 是每次都是 2 个数相乘再除以这次计算两个数的 gcd，而不是前 n 个数的 gcd）；
5. 最后的 gcm 与 lcd 即为所求。

三、参考代码

```
#include<iostream>
using namespace std;
long long gcm(long long, long long);
long long max(long long, long long);
long long min(long long, long long);
int main()
{
    long long a, b;
    long long g, l;
    int n;
    while ( cin >> n )
    {
        cin >> a;
        l = a;
        for ( int i=2; i<=n; i++ )           //步骤 2 与 3 的实现
        {
            cin >> b;
            if ( i == 2 || gcm(a,b)<g )       //步骤 4
                g = gcm(a,b);
            l *= b;
            l /= gcm(a,b);
            a = l;
        }
        cout << g << " " << l << endl;
    }
}

long long gcm(long long a, long long b)     //求最大公约数的函数
{
    long long x, y;
    long long c, d;
    x = max(a,b);
    y = min(a,b);
    while ( x % y != 0 )
    {
        c = x % y;
        d = y;
        x = max(c,d);
        y = min(c,d);
    }
}
```

```
    }
    return y;
}
long long max(long long a, long long b)
{
    if ( a > b )
        return a;
    else
        return b;
}
long long min(long long a, long long b)
{
    if ( a > b )
        return b;
    else
        return a;
}
```

/*其实 max 与 min 函数在 C++标准库头文件<cmath>当中有定义，这里只是本人出于对<cmath>库函数变量为 double 变量的担忧而另外造了个 long long 变量的 max 与 min 函数。在后面的参考代码中可以看到这两个函数用了 3 次。*/

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!

Hint

本题会检查代码，不用函数实现的一律 0 分。

本题是课本原题，见课本 214 页 6.29

二、解题思路

素数判断，这道题是 2014 级 C++ 第一次练习题 E 题(题目链接 <http://acm.buaa.edu.cn/problem/660/>)

的变种，只是加了要求用函数实现这一条，方法不再过多赘述（就是该数与在 $[2, \sqrt{n}]$ 内的任何整数均不能整除即为素数）。

三、参考代码

```
#include<iostream>
#include<cmath>
using namespace std;
bool sushu(long long);
int main()
{
    long long n;
    bool y;

    while ( cin >> n )
    {
        y = sushu(n);
        if ( y == false )
            cout << "jhljx is sangxinbingkuang!" << endl;
        else
            cout << "jhljx is good!" << endl;
    }
}
bool sushu(long long n)           //素数判断
{
    int ghn;
    bool y = true;
    ghn = sqrt(n) + 1;             //这里加了 1 是为了更加严（zhuang）谨（bi）
    for ( int i=2; i<=ghn; i++ )
    {
        if ( n % i == 0 )
        {
            y = false;
        }
    }
}
```

```
        break;
    }
    else
        continue;
}
if ( n == 1 )
    y = false;
if ( n == 2 )
    y = true;
return y;
}
```

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 $n \times n$ 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
```

二、解题思路

//草稿纸有时候能发挥很大作用哦~

这道题我当时在草稿纸上画了四五个棋盘就能发现只要各行各列相隔一点一个小兵即可。另外还发现，

如果 n 为偶数，小兵数目为 $\frac{n}{2} \times n$ ，反之小兵数目为 $\left(\frac{n-1}{2}\right)^2 + \left(\frac{n+1}{2}\right)^2$ 。

三、参考代码

```
#include<iostream>
using namespace std;
long long f(long long);
int main()
{
    long long n;
    long long y;
    while ( cin >> n )
    {
        y = f(n);
        cout << y << endl;
    }
}
long long f(long long n)           //求小兵数目
{
    long long y;
    if ( n % 2 == 0 )
        y = (n/2) * n;
    else
        y = (n/2)*(n/2) + (n/2+1)*(n/2+1);

    return y;
}
```

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 $n, 2n, 3n, 4n, \dots$ 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 $n\%m, 2n\%m, 3n\%m, \dots$)是不是能够得到 $0 \sim m-1$ 之间的所有数, 只有得到了 $0 \sim m-1$ 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 $0 \sim m-1$ 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!

二、解题思路

此题为本次上机题第二难题, 也是我上机期间思考时间最久的题 (最终还是 AC)。

当时思考着这题觉得这题一定不一般，肯定有某种规律在里面，不然做不出来。

当时想出了第一招：如果 m 是 10 的倍数， n 的尾数取某些值时余数不能全部取到 $0 \sim m-1$ 内所有数；后来发现，不对，如果 $n \% m == 0$ 或 $m \% n == 0$ 还是不行；再后来，发现 n 取 4、 m 取 6 时还是不行，而 n 取 3、 m 取 10 的时候是可以的.....

最后一想：如果 n 与 m 互质就可以了， n 或 m 至少有一个取 1 时也行，其他情况统统不行。就这样成了。

三、参考代码

```
#include<iostream>
using namespace std;
long long gcm(long long, long long);
long long max(long long, long long);
long long min(long long, long long);
int main()
{
    long long n, m;
    while ( cin >> n >> m )
    {
        if ( m == 1 || n == 1 )                //特判，如果 m 与 n 至少一个取 1 则 OK
            cout << "jhljxshidadoubi" << endl;
        else if ( gcm(n,m) != 1 )              //n 与 m 不互质
            cout << "shuishuowoshidadoubi" << endl;
        else                                    //n 与 m 互质
            cout << "jhljxshidadoubi" << endl;
    }
}
long long gcm(long long a, long long b)        //事实上就是 A 题的 3 个函数又用了一遍
{
    long long x, y;
    long long c, d;
    x = max(a,b);
    y = min(a,b);
    while ( x % y != 0 )
    {
        c = x % y;
        d = y;
        x = max(c,d);
        y = min(c,d);
    }
    return y;
}
long long max(long long a, long long b)
{
    if ( a > b )
```



```

        return a;
    else
        return b;
}
long long min(long long a, long long b)
{
    if ( a > b )
        return b;
    else
        return a;
}

```

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。($1 \leq n \leq 20$)

Output

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：

Sample 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。

二、解题思路

第四次练习题 A 题的变种，经典递归汉诺塔问题。只是要求多输出次数。次数的话可以发现是 $2^n - 1$ 。

三、参考代码

```
#include<cstdio>
int pf(int);
void h(int,char,char,char);
int main()
{
    int n, coun;
    while(scanf("%d",&n)!=EOF)
    {
        coun = pf(n) - 1;
        printf("%d\n",coun);
        h(n,'A','B','C');
    }
}
void h(int n, char a, char b, char c)
{
    if ( n == 1 )
    {
        printf("%d %c->%c\n",n,a,c);
    }
    else
    {
        h(n-1,a,c,b);
        printf("%d %c->%c\n",n,a,c);
        h(n-1,b,a,c);
    }
}
```

```
}  
int pf(int n)  
{  
    int y = 1;  
    for (int i=n; i>0; i--)  
        y *= 2;  
    return y;  
}
```

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 。

Output

输出斐波那契数列中的第 n 个数。

Sample Input

1
2
3

Sample Output

1
1
2

Hint

这道题用递归和迭代都可以过。纯签到题。

二、解题思路

斐波拉契数列 $\{f(n)\}$ ，就是满足 $f(1)=f(2)=1$ ， $f(n)=f(n-1)+f(n-2)$ 的数列，为教材中的递归示例。

以下为两种做法，都相当简单。

三、参考代码

1.递归做法

```
#include<iostream>
using namespace std;
int feibolaqi(int);
int main()
{
    int n, y;
    while(cin >> n)
    {
        y = feibolaqi(n);
        cout << y << endl;
    }
}
int feibolaqi(int n)
{
    if ( n == 1 || n == 2)
        return 1;
    else
        return feibolaqi(n-1)+feibolaqi(n-2);
}
```

2.迭代做法

```
#include<iostream>
using namespace std;
int main()
{
    int n, x, y, z;
    while(cin >> n)
    {
        x = 1;
        y = 1;
        if ( n == 1 || n == 2 )
            z = 1;
        else
        {
            for ( int i=3; i<=n; i++ )
            {
                z = x + y;
                x = y;
```

```
        y = z;
    }
}
cout << z << endl;
}
```

L 找不到的孩子们

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 $f(2)$ 。 $(0 \leq a,b \leq 10^{18})$

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

二、解题思路

本题是此次上机题最难的题。这题做的时候彻底没思路，上机当时我只知道死算的方法，还各种出 bug，交了还 TLE。

后来参考了王剑锋学长在 QQ 群上分享的做法：

你看那个式子，就是两个数，不断用大的减小的
一直减到两个数循环出现，其实就是其中一个为 0

这就是辗转相减的意思了

但显然不能直接去减，因为范围是 $1e18$ ，肯定会超时的。

注意到，比方说 17 4，相减得到的是 13 4，9 4，5 4，1 4……对吧……

如果辗转相减不行，那相除呢？

$17 \div 4 = 4$ ， $17 \% 4 = 1$

意味着，17 变到 1 的过程中，减了 4 次

减的过程中，4 是不变的，出现的新数字就恰好是 4 个

这样，一步辗转相除其实统计了若干次辗转相减的结果

而辗转相除恰好是 gcd 的过程，复杂度是 \log 的，过了 A 题的应该都会写吧……主要思路就差不多了
注意一点细节和一两个特判就可以了。

其实就是求在辗转相除的过程中每次辗转相除的除数相加。

特判情况为：

1. 前两项有一个为 0，有 2 种数字；
2. 前两项均为 0，只有一种数字。

Get!

三、参考代码

```
#include<iostream>
using namespace std;
long long max(long long, long long);
long long min(long long, long long);
int main()
{
    long long a, b, i;
    long long x, y, c, d;
    while ( cin >> a >> b )
    {
        x = max(a,b);
        y = min(a,b);
        i = 1; //目的是加上 0 这一种数字
        if ( a == 0 && b == 0 ) //特判 2
            i = 1;
        else if ( a == 0 || b == 0 ) //特判 1
            i = 2;
        else
        {
            while ( x % y != 0 )
            {
                i += (x / y);
                c = x % y;
```

```

        d = y;
        x = max(c,d);
        y = min(c,d);
    }
    i += (x / y);
}
cout << i << endl;
}
}
long long max(long long a, long long b)
{
    if ( a > b )
        return a;
    else
        return b;
}
long long min(long long a, long long b)
{
    if ( a > b )
        return b;
    else
        return a;
}

```