# 2014级第五次上机解题报告

巩毅琛 14211072

## 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

解题思路：求两个数的最大公约数用辗转相除法，我们都应该很熟悉。那么a1，a2，a3的最大公约数就是a1和a2的最大公约数与a3的最大公约数，以此类推，a1~an的最大公约数就是a1~an-1的最大公约数与an的最大公约数，于是可以循环使用辗转相除法来完成计算。求n个数的最小公倍数也是用这种思路，其中求两个数的最小公倍数有两个方法。（一），两个数的最小公倍数肯定是较大的那个数的某一个倍数，所以从较大的那个数开始测试其倍数是否是较小数的倍数即可。（二），在求出两个数的最大公约数之后，直接用两数乘积除最大公约数得到最小公倍数，利用a1~an的最小公倍数是a1~an-1的最小公倍数与an的最小公倍数这一规律，循环计算即可。

参考代码：（一）

#include <iostream>

using namespace std;

int f(int a, int b)//求两数最小公倍数

{

int i;

if(a > b)

{

a = a + b;

b = a - b;

a = a - b;

}

for(i = b;; i+=b)

if(i % a == 0&&i%b==0)

{

return i;

break;

}

}

int gcd(int a , int b)//求最大公约数

{

int t;

if (a < b)

{

t = a;

a = b;

b = t;

}

while(b!=0 )

{

t = a % b;

a = b;

b = t;

}

return a;

}

int main()

{

int n;

while(cin>>n)

{

int a[n];

for(int i=0;i<n;i++)

{

cin>>a[i];

}

int zy =gcd(a[0], a[1]);

for(int i=2;i<n;i++)// a1~an的最大公约数就是a1~an-1的最大公约数与an的最大公约数

{

zy =gcd(zy, a[i]);

}

cout<<zy<<" ";

int lcm=f(a[0],a[1]);

for(int i=2;i<n;i++)//a1~an的最小公倍数是a1~an-1的最小公倍数与an的最小公倍数

{

lcm=f(lcm, a[i]);

}

cout<<lcm<<endl;

}

}

（二）

#include <iostream>

using namespace std;

int gcd(int a , int b)//求最大公约数

{

int t;

if (a < b)

{

t = a;

a = b;

b = t;

}

while(b!=0 )

{

t = a % b;

a = b;

b = t;

}

return a;

}

int main()

{

int n;

while(cin>>n)

{

int a[n];

for(int i=0;i<n;i++)

{

cin>>a[i];

}

int zy =gcd(a[0], a[1]);

int lcm= a[0]\*a[1]/zy;

for(int i=2;i<n;i++)

{

zy =gcd(zy, a[i]);

lcm =lcm\*a[i]/gcd(lcm, a[i]);// 在求出两个数的最大公约数之后，直接用两数乘积除最大公约数得到最小公倍数

}

cout<<zy<<" "<<lcm<<endl;

}

}

## 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!

解题思路：判断一个数是否是质数的方法应该都知道（用该数除2,除3，一直除到该数开方加1（开方后可能有被消去的小数部分），若都不能被整除则是素数）。这里就是再练习一下定义函数，实际上我把整个函数都放到了子函数里，主函数只是调用子函数。

参考代码：#include <iostream>

#include <cmath>

using namespace std;

int f(long long a)

{

long long k=sqrt(a)+1;

long long i=2;

if(a==1)

cout<<"jhljx is sangxinbingkuang!"<<endl;

else

{

while(i<=k)

{

if(a%i==0)

break;

i++;

}

if(i>=k)

cout<<"jhljx is good!"<<endl;

else cout<<"jhljx is sangxinbingkuang!"<<endl;

}

}

int main()

{

long long n;

while(cin>>n)

{

f(n);

}

}

## 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

解题思路：画图即可找到规律。如果n为偶数，则小兵即为所有位置的一半，即n\*n/2

|  |  |  |  |
| --- | --- | --- | --- |
| \* |  | \* |  |
|  | \* |  | \* |
| \* |  | \* |  |
|  | \* |  | \* |

如果n为奇数，则小兵即为（所有位置+1）的一半，即（n\*n+1）/2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \* |  | \* |  | \* |
|  | \* |  | \* |  |
| \* |  | \* |  | \* |
|  | \* |  | \* |  |
| \* |  | \* |  | \* |

参考代码：#include <iostream>

using namespace std;

int main()

{

long long n;

while(cin>>n)

{

if(n==1)

{

cout<<"1"<<endl;

}

else

{

if(n%2==0)

cout<< n\*n/2<<endl;

else

cout<< (n\*n+1)/2<<endl;

}

}

}

## 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。。<img:/richtext/res/19/2.jpg>

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!

解题思路：其实就是判断两个数的最大公约数是不是1（判断是否互质），因为如果不是互质，则在用越来越大的n的倍数的时候，总有所谓的“空缺”，n的倍数摸m得不到某个或某些数，于是不能取遍0~m-1之间的所有数，而互质就可以。但其中有特殊情况，若m=1，则甭管n为多少，最大公约数都为1，但此时只能n的倍数摸1只能取到0，没有1，所以输出shuishuowoshidadoubi。若n=1且m≠1则一定能取到所有的数，想想就知道。然后在最后n≠1且m≠1的时候就用一开始说的那思路吧。

参考代码：#include <iostream>

using namespace std;

int f(int a,int b)

{

int t;

if (a < b)

{

t = a;

a = b;

b = t;

}

while(b!=0 )

{

t = a % b;

a = b;

b = t;

}

return a;

}

int main()

{

int n, m;

while(cin>>n>>m)

{

if(m==1)

cout<<"shuishuowoshidadoubi"<<endl;

else if(n==1&&m!=1)

cout<<"jhljxshidadoubi"<<endl;

else if(n!=1&&m!=1)

{

if(f(n, m)==1)

cout<<"jhljxshidadoubi"<<endl;

else cout<<"shuishuowoshidadoubi"<<endl;

}

}

}

## 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。**

解题思路：对于输出移动的具体方案，在练习赛中我们都练过了，解题报告中我也写的很详细，我直接复制粘贴上回我写的吧

（这是用递归来实现的

先注意下，在这定义的递归函数里，移动的方向都是第一个到第三个，不论第一个第二个第三个代表的字符分别是什么，总是这么空出第二个来从第一个移动到第三个，这是在汉诺塔移动所有盘子的共通的地方  
要把n个从A通过B移到C，就先要把上面n-1个从A移到B【1】，然后把第n个移到C。在把n-1个从B移动到C【2】。（一）  
那么首先要解决【1】这个问题，也是用同样的方法，和方才的大问题（一）一样的问题，所以相同的思路，只是把B和C交换位置而已（满足从第一个到第三个移动的共通点），也就是说把n-2个移到C，方法用的和【1】一样，就这样递归实现。解决【2】时也同思路，把A和B互换（还是满足满足从第一个到第三个移动的共通点），因为要将位于B的n-1个移到已经有第n个的C上。然后就这么递归下去，也甭管n不断减1后n变得很小时盘子是怎么移动的，根据写的递归函数他自己就会算出来，没必要想那么多。

）

然后对于输出需要移动的步数，拿n=3的时候举例子，n=3的前三步就是重复n=2时候，把小的两块移动到另一个一个柱子。然后就是将最大块移动到另一块柱子上，在继续重复n=2把小的两块移动到另一个（指存在3的一个）柱子上。扩展到n，即f（n）=2\*f（n-1）+1。

参考代码：

#include <iostream>

#include <cstdio>

using namespace std;

int main()

{

void hanoi(int ,char ,char ,char );

int val (int);

int m;

while(cin>>m)

{

printf("%d\n",val(m));

hanoi(m,'A','B','C');

}

}

void hanoi(int n,char one,char two,char three)

{

if (n == 1)

{

printf("%d %c->%c\n",n , one, three);

}

else

{

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

printf("%d %c->%c\n", n, one ,three);

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

}

}

int val(int n)

{

int c;

if(n==1)

c=1;

else c=2\*val(n-1)+1;

return c;

}

## 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

解题思路：思路很简单，不必多说。f(n)=f(n-1)+f(n-2)。详情可见c++课本199页。

参考代码：

#include <iostream>

using namespace std;

int f(int n)

{

if(n<=1)

return n;

else

return f(n-1)+f(n-2);

}

int main()

{

int n;

while(cin>>n)

{

cout<<f(n)<<endl;

}

}

## G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<=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

解题思路：想几个例子在纸上多演算即便找到规律。直接说规律不好说。我就拿17和4举个例子。（特殊情况a=0且b=0则只求一个数，a和b有一个为0则有两个数，想想就知道）。对于17和4，大小前后顺序不用管，若17在前，则是17 4 13 9 4 5 1 4 3 1 2 1 1 0，若4在前，则是4 17 13 4 9 5 4 1 3 2 1 1 0。从中可以看出，首先是17不断减4,到13 到9 到5 ，减到最后保证比4大。减了3次，加上原来的17，那么就有了4个数（a/b的整数除法得到），之后再用4（这是17减的那个4，不是a/b得到的那个4）来不断减1（a%b得到的），减到3，到2，到1，到0，减了4次（新a（原b）/新b（原a%原b）整数除法得到）那么就又有了四个数，再加上原来的4就是5个，所以一共9个数。实际上出现0就意味着结束了（新a可以被新b整除），拿刚刚的到1，到0时的情况举例子，减到最后就是1 0 1 1 0 1 1 0 1 1 0····如此循环始终是0和1于是没有了继续讨论的意义。

参考代码：

#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<b)

{

long long t=b;

b=a;

a=t;

}

long long c=a/b;

long long m=c;

while(a%b!=0)

{

long long d=a%b;

a=b;

b=d;

c=a/b;

m+=c;

}

cout<<m+1<<endl;

}

}

}