# 2014级C++第五次上机

## 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：

可以考虑先求前两个数的最大公约数和最小公倍数，然后输入第3个数，再求第三个数和最大公约数的最大公约数，和最小公倍数的最小公倍数，然后依次循环。

（其中由于求最大公约数和最小公倍数的多次使用，我定义了2个函数，然后在main函数中调用）；（对于产生了Runtime Error (SIGFPE) – 0的可能是没判断整除的时候到时gcd没有得出（我加了注释的那段））；（我是直接改输入函数第一个数的值，故用了&符号）

## 参考代码：

#include<iostream>

using namespace std;

int gcd(int &x,int &y)//求公约数函数

{

int a=x,b=y;

if(a<b)

{

int z=a;

a=b;

b=z;

}

int gcd;

if(a%b==0) //要判断整除

gcd=b;

else{

while(a%b!=0)

{

gcd=a%b;

a=b;

b=gcd;

}

}

x=gcd; //是x为所求的值

}

int lcm(int &x,int &y)//求最小公倍数函数

{

int a=x;

gcd(x,y);

int lcm=a\*y/x;

x=lcm;

}

int main()

{

int n;

while(cin>>n)

{

int a,b,c=0;

cin>>a>>b;

int x=a;

gcd(x,b); //得到最大公约数x；

lcm(a,b); //得到最小公倍数a；

for(int i=0;i<n-2;i++)

{

cin>>c;

gcd(x,c); //上一个最大公约数和c求最大公约数

lcm(a,c); //上一个最小公倍数和c求最小公倍数

}

cout<<x<<" "<<a<<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!

## 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**

## 解题思路：

主要还是函数的调用。可以用void定义函数来直接在函数中输出结果（这样比较方便）。然后就是函数里面设置了，求素数可以从2~sqrt(n)之间的值对n求模，若都不能整除，则为素数。另外再考虑一下n为1的特殊情况。

## 参考代码：

#include<iostream>

#include<cmath>

using namespace std;

void zhishu(long long n) //求素数的函数

{

int c=0;

if(n==1)cout<<"jhljx is sangxinbingkuang!"<<endl; //特殊情况

else{

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

{

if(n%i==0)c++;

}

if(c!=0)cout<<"jhljx is sangxinbingkuang!"<<endl;

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

}

}

int main()

{

long long n;

while(cin>>n)

zhishu(n); //调用

}

## 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。

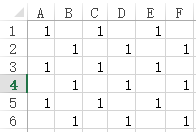
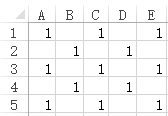
## Sample Input

1  
2  
3

## Sample Output

1  
2  
5

## 解题思路：



从上面是不是发现了什么~

可以看到5\*5时为 3\*3+2\*2,6\*6时为3\*3+3\*3；所有我们可以知道为偶数的时候答案为((n/2)^2)\*2，奇数时为(n/2)^2+((n+1)/2)^2。需要注意的是，答案会超出int的范围，所有可以定义为long long类型。

## 参考代码：

#include<iostream>

using namespace std;

int main()

{

long long n,a,b,ans;

while(cin>>n)

{

a=(n+1)/2;

b=n/2;

ans=a\*a+b\*b;

cout<<ans<<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/688/2.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=**2**;

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：

多次试验一下就知道如何n,m最大公约数不为1是就不会取遍0~m-1之间的所有数，这是为什么呢？

如上面的hint，你会发现（15+3）%5，（15+6）%5，（15+9）%5，……会开始重复一遍前面的值，若m,n互质的话，他会到m次就会重复（n<m时要m+1次），m次中没有重复，所有就会取遍0~m-1之间的所有数。若m,n不互质，那么他就会到他的最大公约数时重复，所有无法取遍0~m-1之间的所有数。

所以，问题就化为简单求n,m是否互质，即求m,n的最大公约数是否为1。

## 参考代码：

#include<iostream>

using namespace std;

int main()

{

int n,m;

while(cin>>n>>m)

{

int a=n,b=m;

if(a<b)

{

int z=a;

a=b;

b=z;

}

int gcd;

if(a%b==0)

gcd=b;

else

{

while(a%b!=0)

{

gcd=a%b;

a=b;

b=gcd;

}

}

if(gcd==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

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

## 解题思路：

最基本的情况是移动一个盘子，所以，递归结束的地方就是只移动一个盘子的时候。

然后是移动n个盘子，要让n个盘子从A移动到C，得先把第n个盘子上面的 n-1个盘子搬到B，然后把第 n个搬到C，然后把B上的所有盘子搬到C。

要把 n-1 个盘子从 A 搬到 B，得先把第 n-1 个盘子上面的 n-2 个盘子搬到 C，然后把第

n-1个盘子搬到B，然后把 C上的所有盘子搬到B。

按照这样的想法，n就在逐步减小，最后变成基本情况，递归结束。

## 参考代码：

#include<iostream>

#include<cstdio>

#include<cmath>

using namespace std;

void fuc(int n,char a,char b,char c)

{

if(n==1)

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

else

{

fuc(n-1,a,c,b);

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

fuc(n-1,b,a,c);

}

}

int main()

{

int n;

while(~scanf("%d",&n)){

int sum=pow(2,(double)n)-1;

printf("%d\n",sum);

fuc(n,'A','B','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

## Hint

这道题用递归和迭代都可以过。纯签到题。

## 解题思路：

首先最基本情况是n=0时为0，n=1时为1，然后当n>2时就是前一个与前二个相加，即f(n)=f(n-1)+f(n-2)。

## 参考代码：

#include<iostream>

#include<cstdio>

using namespace std;

int fuc(int n)

{

if(n==0||n==1)

return n;

else

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

}

int main()

{

int n;

while(cin>>n)

{

cout<<fuc(n)<<endl;

}

}