# 14级第五次上机解题报告

Ajhljx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

分析：先输入两个数，计算它们的最大公约数，最小公倍数，然后接着输入下面的数，同样进行这样的计算；最后就可以得到结果。

参考代码：

#include <iostream>  
#include <cmath>  
using namespace std;  
  
int gcd(int a, int b)  
{  
    if(a < b)

 swap(a, b);  
    while(b > 0)  
    {  
            a = a % b;  
            swap(a, b); //swap交换a与b的值；         
    }  
    return a;  
}  
  
int main()  
{  
    int a, b, c;  
    int n, i;  
    while(cin >> n)  
    {  
              cin >> a >> b;  
              c = a \* b / gcd(a, b);

     a = gcd(a, b);  
              for(i = 2; i < n; i ++)  
              {  
                      cin >> b;

      a = gcd(a, b);  
                      c = c \* b / gcd(c, b);  
              }  
           
              //c /= a;  
  
              cout << a << " " << c << 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**

分析：这道道用bool应该也是可以的；

参考代码：

#include<iostream>

#include<cmath>

#include<cstdio>

using namespace std;

bool PrimeNum(long long x)// PrimeNum素数

{

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

{

if(x%i==0)

return false;

}

return true;

}

int main()

{

long long t ;

while(cin>>t)

{

if(t==1)

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

else

{

bool flag = PrimeNum(t);

if(NumPrime(t)==true)

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

else

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

}

}

}

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

分析：分奇偶情况；(1)当n为偶数时画图可知结果为(n\*n)/2;

(2)当n为奇数时同样可以得到公式int t=(n+1)/2;结果为t\*t+(t-1)\*(t-1);

注意：题目中虽然说n在int范围内，但是n\*n就会溢出；所以要用long 或long long 从始到终就没发现这个。。。。

参考代码：

#include<iostream>

using namespace std;

int main()

{

long long n;

while(cin>>n)

{

long long ans=0,t=0;

if(n%2==0)

{

ans=(n\*n)/2;

cout<<ans<<endl;

}

else

{

t=(n+1)/2;

ans=t\*t+(t-1)\*(t-1);

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!

分析：

就是判断看它们是不是互质

参考代码：

#include<cstdio>

#include<iostream>

using namespace std;

int main()

{

int a,b,temp;

while(scanf("%d%d",&a,&b)!=EOF)

{

while(b!=0)

{

temp=b;

b=a%b;

a=temp;

}

if(1==a)

printf("jhljxshidadoubi\n");

else

printf("shuishuowoshidadoubi\n");

}

}

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<=10)

# 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。**

分析：在下面大家应该都做过练习题了吧，对的它就是A题

开始递归。。。

参考代码：

#include<iostream>

#include<cstdio>

#include<cmath>

using namespace std;

int fun(int ,char,char,char );

int main()

{

int n;

char A='A',B='B',C='C';

while(scanf("%d",&n)!=EOF)

{

cout<<pow(2,double(n))-1<<endl;

fun(n,A,B,C);

}

}

int fun(int n,char a,char b,char c)

{

if(1==n)

{

cout<<n<<" "<<a<<"->"<<c<<endl;

}

else

{

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

cout<<n<<" "<<a<<"->"<<c<<endl;

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

}

}

Fjhljx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。

分析：斐波那契数列：接着递归吧。。。

参考代码：

#include<iostream>

#include<cstdio>

using namespace std;

int fun(long long);

int main()

{

int n;

while(scanf("%lld",&n)!=EOF)

{

cout<<fun(n)<<endl;

}

}

int fun(long long n)

{

if(1==n)

return n=1;

else if(2==n)

return n=1;

else if(n>=3)

return n=fun(n-1)+fun(n-2);

}

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

分析：

参考代码: