

A. jhljx 学 gcd

题目描述

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
```

解题思路

最大公约数的求法为辗转相除法(前面题里已经见过), N 个数的最大公约数即在输入第三个数的时候再计算前两个数的最大公约数和第三个数的最大公约数;两个数的最小公倍数的求法为两数相乘除以二者的最大公约数,输入第三个数的时候计算前两个数的最小公倍数和第三个数的最小公倍数即可。依次类推即可求出 N 个数的最大公约数和最小公倍数,建议使用构造函数来使代码看着简便。

参考代码

```
#include<iostream>
#include<cstdio>
#include<iomanip>
#include<cstdlib>
using namespace std;
int num,n;
int gcd(int , int );
int lcm(int , int );
//int min,max;
int q,w;
int main()
{ int min,max;
```

```

while( cin >> n )
{
    min = 0;
    max = 1;
    cin >> q >> w;
    min = gcd(q,w);
    max = lcm(q,w);
    for( int i = 3 ; i <= n ; i++ )
        {cin >> num;
        min = gcd(num,min);    //求出所有数的最大公约数
        max = lcm(num,max);    //求出所有数的最小公倍数
        }
    cout << min << " " << max << endl;
    max = 0;
    min = 0;
}
}

int gcd(int a, int b)    //求两数最大公约数的函数
{
    if (a == 0)
        return b;
    if (b == 0)
        return a;
    if (a < b)
    {
        int tmp = a;
        a = b;
        b = tmp;
    }
    while (b != 0)
    {
        int tmp = a % b;
        a = b;
        b = tmp;
    }
    return a;
}

int lcm (int a, int b)    //求两数最小公倍数的函数
{
    return a*b/gcd(a,b);
}

```

B. jhljx 学素数

题目描述

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

判断一个数是否为素数的代码前面考过了，这里不详

细分析了。本题主要就是考察对函数的应用，输入一个数，让通过函数的判断来看是否为素数，使程序可读性更强，更加条理。

参考代码

```
#include<iostream>
#include<cstdio>
#include<iomanip>
#include<cmath>
using namespace std;
int pd(long long );
int main()
{long long n;
    while(cin >> n)
        if (pd(n) == 1 )
            cout <<"jhljx is good!"<<endl;
        else
            cout << "jhljx is sangxinbingkuang!"<<endl;
}
int pd(long long m)    //构造一个名为判断的函数来判断是否为素数
{
    int ans = 0;
    int i;
    if (m == 1 ) ans = 0 ;
    else
    {double sqrtm = sqrt( m ) ;
    for ( i = 2; i <= sqrtm ; i++ )
        if ( m % i == 0 ) break;
        if ( sqrtm < i ) ans = 1;
        else ans = 0;}
    return ans;
}
```

C. jhljx 学下棋

题目描述

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
```

解题思路

本题题意就跟涂黑白相间颜色的天花板似的，看你在网格里能涂黑色的个数。 $N \times n$ 的方格，如果是偶数个格子（即 n 为偶数）的话，明显一半涂黑一半涂白即可；如果是奇数个格子的话，需要 $(n \times n + 1) / 2$ 个格子涂黑已达到最大。

参考代码

```
#include<iostream>

using namespace std;

long long n,ans;

int main()
{
    while (cin>>n)
    {
        ans=(n/2)*n+(n+1)*(n%2)/2; //把两句判断结果变为一句写了而已
        cout<<ans<<endl;
    }
}
```

D. jhljx 的强迫症

题目描述

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。。欸。。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

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

本题初步摸索会发现原题可以转化为求 n , m 的是否互质的问题。

因为要使 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, 则 $n*1$ 到 $n*m$ 最后模 m 的值分别对应 0 到 $m-1$, 才能说明满足条件。

如果 n 和 m 不互质, 如 3 和 6, 则会导致 $3*2$ 和 $3*6$ 模 6 的结果相同, 使得不能取尽 1,2,3,4,5 来满足题意。

而求互质的方法即判断其最大公约数是否为 1 即可。

参考代码

```
#include<iostream>
#include<cmath>
#include<iomanip>

using namespace std;
int gcd(int, int);
int main()
{ int n, m;
  while(cin>>n>>m)
    {if(gcd(n, m)==1) cout<<"jhljxshidadoubi"<<endl;
     else  cout<<"shuishuowoshidadoubi"<<endl;}}

int gcd(int a, int b)
{int c, i=0;
 for (;a%2==0&&b%2==0;i++)
 {a/=2;b/=2;}

while (a!=b)
{if (a<b)
 {a=a+b;
 b=a-b;
 a=a-b;}}
 a-=b;}
 while(i-->0) a*=2;
 c=a;
 return c;
}
```

E. 汉诺塔 再度来袭

题目描述

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)

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:

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

汉诺塔的问题已经考过两次,本次依旧没有太大变动,是修改了输出格式并且要求输出总共需要移动的步数。根据数列递推项求和即可知道,输如 N 需要的步数为 2^n-1 , 所以算出来,再调整下输出格式即可。

参考代码

```
#include <iostream>
#include<cmath>
using namespace std;
int bu;
void towers( int, char, char, char );
int nDisks;
char A,B,C;
int main()
{
    while(cin >> nDisks)
    {
        int bu = 1;
        for(int i = 1 ;i<=nDisks ; i++)
            bu=bu*2;
        bu = bu-1;
        cout <<bu<<endl;
        towers( nDisks, 'A', 'C', 'B' );}
}

void towers( int disks, char start, char end, char temp )
{
    if ( disks == 1 )
        {cout << "1 " << start << "->" << end << '\n';}
    else
    {
        towers( disks - 1, start, temp, end );
        cout << disks<<" " << start << "->" << end << '\n';
        towers( disks - 1, temp, end, start );
    }
}
```

F. jhljx 学斐波那契数列(I)

题目描述

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。

解题思路

本次上机的水题，主要是对递归的熟悉和练习，直接利用书上的原递归斐波那契数列的输出方法输出即可。

参考代码

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

int fib(int n )
{if (n==1||n==2)
  return 1;
  else return fib(n-1)+ fib(n-2);
}
```

