

2014 级 C++ 第五次上机解题报告

14211066 宋晏桢

对上机真是充满了深深的恶意啊==各种马虎检查不出来==。

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
```

解题思路

反正我是一看到这道题就开始去百度了 2333，首先我们要输入 n 个数，这就需要开数组了。至于怎么求 n 个数的最大公约数呢，我们不妨依次考虑先求两个数的最大公约数，然后再求这个最大公约数与下一个数的最大公约数，以此类推，我们便可以得到 n 个数的最大公约数。其次，求最小公倍数的过程也与此类似，先求前两个数的最小公倍数，再求这个最小公倍数与下一个数的最小公倍数，以此类推，你懂得.....至于怎么求最小公倍数，我才不会告诉你就是把两个数相乘再除以它们两个的最大公约数的。

为了方便，我们可以在代码中写一个函数来求最大公约数。

参考代码

```
#include <iostream>
using namespace std;
int gys(int a, int b)//定义函数 gys 来求两个数的最大公约数
{
    int c=1, gcd;
    while(c>0)
    {
        c=a%b; a=b; b=c;
    }
    gcd=a;
    return gcd;
}
//函数中运用了更相减损的方法，此处不必再过多阐述了吧？
int main()
{
    int n;//n 为输入数字的个数
    while(cin>>n)
    {
        int sb[n+1];//定义一个数组，其中有 n+1 个元素，用 1~n 分别表示 1~n 个数
        for(int t=1; t<=n; t++)
        {
            cin>>sb[t];//输入 n 个数字
        }
        int p=gys(sb[1],sb[2]);//取前两个数的最大公约数
        for(int i=2; i<=n; i++)
        {
            p=gys(p,sb[i]);//分别计算前两个数的最大公约数和下一个数的最大公约数，以此类推，便可求得 n 个数共同的最大公约数
        }
        int q=sb[1]*sb[2]/gys(sb[1],sb[2]);//取前两个数的最小公倍数
        for(int j=2; j<=n; j++)
```

```

        {
            q=q*sb[j]/gys(q,sb[j]);//分别计算前两个数的最小公倍数和下一个数的最小公
倍数，以此类推，便可求得 n 个数共同的最小公倍数
        }
        cout<<p<<" "<<q<<endl;//输出即可
    }
}

```

Ljx 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

由于这道题需要我们用函数来解决，所以我们只需要写一个判断一个数是否为质数的函数即可，至于怎么求呢，只需要将这个数 n 分别模从 2 一直到根号 n ，如果在这个区间内模的结果不为 0，则这个数便是质数。此外，还需注意 1 和 2 要分别考虑，其中 1 不是质数，2 是质数。

参考代码

```
#include <iostream>
#include <cmath>
using namespace std;
void sb(long long x) //定义函数判断一个数是否为质数
{
    int p=0;
    double y; y=sqrt(x); //其中 y 需要 double，因为 y 是 x 开平方
    if(x==1) cout<<"jhljx is sangxinbingkuang!"<<endl;
    else if(x==2) cout<<"jhljx is good!"<<endl; //此处对 1 和 2 进行单独
判断
    else
    {
        for(int i=2; i<=y; i++)
        {
            if(x%i==0) p=1;
        }
        if(p==1) cout<<"jhljx is sangxinbingkuang!"<<endl;
        else cout<<"jhljx is good!"<<endl;
    } //当 x 大于 2 时，用常规方法判断即可，并直接输出
}
int main()
{
    long long n;
    while(cin>>n)
    {
        sb(n); //这里只要无脑写上函数就好啦！
    }
}
```

```
}
```

Ljx 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
```

解题思路

由于刚看到这道题时没有任何思路，我们不妨先列举几个例子：

1-1,2-2,3-5,4-6,5-13,6-18,7-25.....

于是我们对于如何放置才能最多也应该有了一定的认识，当 n 为奇数是，棋子按照对角线依次向下向上排列，表达式便可又等差数列求和公式得出 $ans=(n/2)*(n/2)+(n+1)/2*(n+1)/2$ 。当 n 为奇数时，棋子按照中间类似于对角线的线向上向下排列，表达式可得出 $ans=n*n/2$ 。

此外，这道题的 n 一定要用 `long long`，至于为什么呢，我个人认为是 $n*n$ 会超出 `int` 的范围。否则用 `int` 的话真的会欲哭无泪的（比如我）

参考代码

```
#include <iostream>
using namespace std;
int main()
{
    long long n, ans; //一定要用 long long 哦~
    while(cin>>n)
    {
        if(n%2==1) //判断当 n 为奇数时
        {
            ans=(n/2)*(n/2)+(n+1)/2*(n+1)/2;
        }
        else if(n%2==0) //判断当 n 为偶数时
            ans=n*n/2;
        cout<<ans<<endl;
    }
}
```

ljx 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。。欸。。一天，树荫姐给了 jhljx 两个数 n 和 m ，树荫姐说我们来求 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。

$3+0=3, 3\%5=2$;

$3+3=6, 6\%5=1$;

$3+3+3=9, 9\%5=4$;

$3+3+3+3=12, 12\%5=2$;

$3+3+3+3+3=15, 15\%5=0$;

$3+3+3+3+3+3=18, 18\%5=3$;

$m=5$, 这些余数取到了 0, 1, 2, 3, 4, 满足! get!

解题思路

据说这是一个数论问题，我们不妨这么想，题目中并没有给出 n 最大的倍数，所以我们应该采取分别判断的方式，再想如果 n 和 m 不互质，那么不管 n 的几倍模 m 一定无法全部取到 $m-1$ 以内的数，相反如何 n, m 互质，那么 n 的倍数模 m 一定会得到 $m-1$ 内的所有的数。

（自行理解吧，我也只能理解到这了）所以我们只需判断 n 和 m 是否互质即可，即判断二者最大公约数是否为 1，其中 $n=1$ 或 $m=1$ 的情况也要单独考虑，此时是一定成立的。

参考代码

```
#include<iostream>
using namespace std;
int main()
{
    int n, m;
    while(cin>>n>>m)
    {
        int c=1, gcd;
        if(m==1 || n==1)//判断当 n 等于 1 或 m 等于 1 时的特殊情况，此时题中要求成立
            cout<<"jhljxshidadoubi"<<endl;
        else//这里就开始求最大公约数啦
        {
            while(c>0)
            {
                c=n%m; n=m; m=c;
            }
            gcd=n;
        }
        if(gcd==1)//如果最大公约数为 1，那么 n 和 m 互质，成立
            cout<<"jhljxshidadoubi"<<endl;
        else if(gcd>1)//如果不为 1，则 n 和 m 不互质，不成立
            cout<<"shuishuowoshidadoubi"<<endl;
    }
}
```


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：

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

本题和练习赛的那道汉诺塔的题目如出一辙，只是改变了最终要挪到的位置并需要输出所用的步数而已。仍然采用递归的思想，先把前 $n-1$ 个盘子挪到中间，再把最后一个挪到第三个柱子，最后把中间的 $n-1$ 个盘子挪到第三个柱子即可。

此外，至于步数的问题，通过找规律可以发现，步数 $= n-1$ 个盘子时步数 $* 2 + 1$ ，于是这个我们可以用一个 **for** 循环来解决。

参考代码

```
#include <iostream>
using namespace std;

void sb(int n, char a, char b, char c) //递归函数
{
    if(n==1) cout<<n<<" "<<a<<"-"<<c<<endl; //当剩余一个盘子时，将其移到第三个柱子上
    else
    {
        sb(n-1, a, c, b); //将前 n-1 个盘子移到中间
        cout<<n<<" "<<a<<"-"<<c<<endl; //将第 n 个移到第三个柱子
        sb(n-1, b, a, c); //将前 n-1 个盘子移到第三个柱子
    }
}

int main()
{
    int n, s;
    char a='A', b='B', c='C';
    while(cin>>n)
    {
        int x=0;
        for(int i=1; i<=n; i++)
            x=2*x+1; //计算步数，并输出
        cout<<x<<endl;
        sb(n, a, b, c); //直接运用递归函数无脑输出即可
    }
}
```

ljx 学斐波那契数列(I)

Problem Description

jljljx 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

解题思路

签到题，一个简单的递归即可搞定，课本上都有的，不需要多说了吧.....

参考代码

```
#include <iostream>
using namespace std;
int sb(int n)//定义函数，求第 n 项斐波那契数列
{
    if(n==0 || n==1)
        return n;
    else
    {
        return sb(n-1)+sb(n-2);
    }
}
```

```
int main()
{
    int n;
    while(cin>>n)//输入 n，代表第 n 项
    {
        cout<<sb(n)<<endl;//无脑输出即可
    }
}
```

找不到的孩子们

Orz 又是一道用心理测量者命名的题目，对于我等弱渣真的无法理解，就这样吧。。TT