

第五次上机解题报告

By——14211103 田旺

写在前面：

个人来说，感觉这次上机充分体现了平日刷题的重要性。因为这次上机题好多都是可以从前出过的练习赛或者上机题中找到影子的。可是，我这次上机的成绩还是不理想。7 道题只过了 3 道。而上机结束后，我回到寝室就做完了剩余的 3 道（还有一道貌似不在我的能力范围内）。上机的心态还是太紧张吧。造成紧张的原因我想还是平日刷题不够，信心不足所致。既然如此，还有什么好说的呢？滚去刷题

A 题：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：

这道题的题意很简单，就是求多个数的最大公约数和最小公倍数。在求多个数的最大公约数和最小公倍数时，采取的方法是先求前两个数的 gcd 或 lcm，再求前两个数的 gcd 或 lcm 和第三个数的 gcd 或 lcm。在本题中，我采取的是一边输入数据，同时进行计算 gcd 和 lcm 的方法，避免了使用数组（当然用数组会更简单）。

参考代码：

```
#include<iostream>
using namespace std;
int zdgy(int a,int b)
{
    int c;
    while(b%a!=0)
    {
        c=b%a;
        b=a;a=c;
    }
    return a;
}
//定义求两个数最大公约数的函数（用的是辗转相除法）
int zxgb(int a,int b)
{
    int c;int x=a*b;
    while(b%a!=0)
    {
        c=b%a;
        b=a;a=c;
    }
    return x/a;
}
//定义求两个数最小公倍数的函数，用的是最大公约数和最小公倍数的关系
int main()
{
    int n;int a,b,c,d;//n 为输入的数据数，a b d 为输入的数，c 用于记录最大公约数
    while(cin>>n)
    {
        if(n==2)
        {
            cin>>a>>b;
            cout<<zdgy(a,b)<<" "<<zxgb(a,b)<<endl;
        }//如果只有两个数，需单独算。
        if(n>=3)
        {
            cin>>a>>b;c=zdgy(a,b);int s=zxgb(a,b);//由于后面的计算需要用到最初
            的 2 个数的 lcm 和 gcd，所以先计算前面两个数的。并记录下来，作为后面的计算基础。
```

```

for(int i=3;i<=n;i++)//因为前面已经输入了两个数,所以应该再输入 n-3 个数。
{
    cin>>d;//读入输入的数
    zdgy(c,d);//计算新读入的数与前两个数的 gcd 的 gcd
    c=zdgy(c,d);//将这个新的 gcd 赋给 c, 参加下一次与新输入的数的计算。

    zxgb(s,d);
    s=zxgb(s,d);//计算最小公倍数, 逻辑同上。
}
cout<<c<<" "<<s<<endl;//按格式输出
}
}
}

```

B 题: 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：

这道题以前是出过的，就是第一次练习赛中的 E 题“判断质数”，那么，剩下的就是需要写一个函数的问题了，因为不用函数就不给分呢 我采取的办法就是直接把原来 main 函数中的代码写在另一个自己定义的函数里。实质上是没有变化的，只是多了个函数的形式而已。

参考代码：

```
#include<iostream>
#include<cmath>
using namespace std;
long long sushu(long long); //定义一个判断素数的函数，为了满足题目的数据要求，我把它的变量定义为 long long 型。
int main()
{
    long long n;
    while(cin>>n)
    {
        sushu(n); //此处调用 sushu 函数，其实就相当于将原来 main 函数里代码全部写到了 sushu 函数里。
    }
}
long long sushu(long long a) //定义 sushu 函数。判断一个数是不是素数，就是要看在【1，n】之间，会不会有 n 的约数。
{
    int i,c;
    if(a==1)
        cout<<"jhljx is sangxinbingkuang!"<<endl;
    else
    {
        for(i=sqrt(a);i>0;i--) //注意，此处只需要从一个数的平方根处开始挨个检验就好，而不是从该数开始。这样可以减少程序运行时间。避免超时。
        {
            c=a%i;
            if(c==0&& i==1)
                cout<<"jhljx is good!"<<endl;
            else if(c==0&& i>1&& i<a) //表明在中间找到了该数的约数
                {cout<<"jhljx is sangxinbingkuang!"<<endl; break;}
        }
    }
}
```

C 题: 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 $n \times n$ 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：

这道题嘛，其实自己多画几个格子摆一下就可以发现规律了。那么，我们可以看出，所求结果就是一个等差数列的求和，只不过求和的项数是和 n 有关的。实质上，这道题就是一个简单的计算而已。

参考代码：

```
#include<iostream>
using namespace std;
int main()
{
```

```
    long long n;//注意，这儿是坑点之一！虽然题目中说保证  $n$  在 int 范围内，但实际上，
```

由于后面会用到平方运算，会导致其超出 int 范围，所以，还是声明为 long long 型吧。

```
while(cin>>n)
{
    if(n%2==0)//判断 n 的奇偶性，并按照之前找出的规律计算就好。
        cout<<(n/2)*(n/2)*2<<endl;
    if(n%2!=0)
        cout<<((n-1)/2)*((n-1)/2)*2+n<<endl;
}
```

D 题: 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。。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

```
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:

坑题!!! 允许我先吐槽一下, 本人实在没发现什么取模运算的规律 后来说, 原来只要这两个数互质就能满足题意啊 好吧, 既然如此, 那我们就来判断两个数是否互质和好了。我采取的办法是, 如果这两个数的乘积等于它们的最小公倍数, 那么这两个数就是互质的。那么问题来了, 怎么求最小公倍数呢? 这个问题, 做过很多遍了好吗? !

参考代码:

```
#include<iostream>
using namespace std;
int zxgb(int a,int b)
{
    int c;int x=a*b;
    while(b%a!=0)
    {
        c=b%a;
        b=a;a=c;
    }
    return x/a;
} //定义计算最小公倍数的函数。
int main()
{
    int a,b;
    while(cin>>a>>b)
    {
        if(a*b==zxgb(a,b)) //判断 ab 的乘积与它们的最小公倍数是否相等。
            cout<<"jhljxshidadoubi"<<endl;
```

```
else
    cout<<"shuishuowoshidadoubi"<<endl;
}
```

E 题：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：

又见汉诺塔，这道题不光是松油老师上课讲过，还在前面的练习赛中出现过，应该没什么要说的了。唯一不同的就是需要多输出一个总共的次数而已。而这个总次数就是 $2^n - 1$ 。直接计算并输出就好。

参考代码：

```
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;
void hannuo(int,char,char,char);
```

```

long long times;
int main()
{
    int n;int ans;
    char a='A',b='B',c='C';
    while(~scanf("%d",&n))
    {
        ans=pow(2.0,n)-1;//主要是这里需要多输出一个总次数而已，其它的都是按照松
油老师的递归思路实现的。
        printf("%d\n",ans);
        hannuo(n,a,b,c);
    }
}
void hannuo(int num,char x,char y,char z)//汉诺塔的递归函数。
{
    if(num==1)
    {
        printf("%d %c->%c\n",num,x,z);
    }
    else
    {
        hannuo(num-1,x,z,y);
        printf("%d %c->%c\n",num,x,z);
        hannuo(num-1,y,x,z);
    }
}

```

F 题： 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：

嗯，Hint 里不是说过了吗，纯签到题 · · · · · 就用书上的代码稍微修改一下就可以过了。

参考代码：

```
#include<iostream>
#include<cstring>
using namespace std;
unsigned long fibonacci(unsigned long);
int main()
{
    int n;
    while(cin>>n)
    {
        cout<<fibonacci(n)<<endl;
    }
}
unsigned long fibonacci(unsigned long num)//定义斐波那契的递归数列
{
    if((num==0) || (num==1))
        return num;
    else
        return fibonacci(num-1)+fibonacci(num-2);
}
```

G 题：找不到的孩子们

真心不会 ORZ.....