Assignment: 数据成分应用练习

You have not submitted. You must earn 80/100 points to pass.

1. [**Instructions**](https://www.coursera.org/learn/jisuanji-biancheng/programming/nXnUt/shu-ju-cheng-fen-ying-yong-lian-xi)
2. [My submission](https://www.coursera.org/learn/jisuanji-biancheng/programming/nXnUt/shu-ju-cheng-fen-ying-yong-lian-xi/submission)
3. [Discussions](https://www.coursera.org/learn/jisuanji-biancheng/programming/nXnUt/shu-ju-cheng-fen-ying-yong-lian-xi/discussions)

抄写题＃1: 约瑟夫问题

[来源: POJ](http://pkuic.openjudge.cn/ziyoulianxi/15/) (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

**总时间限制: 1000ms 内存限制: 65536kB**

描述

约瑟夫问题：有ｎ只猴子，按顺时针方向围成一圈选大王（编号从１到ｎ），从第１号开始报数，一直数到ｍ，数到ｍ的猴子退出圈外，剩下的猴子再接着从1开始报数。就这样，直到圈内只剩下一只猴子时，这个猴子就是猴王，编程求输入ｎ，ｍ后，输出最后猴王的编号。

输入

每行是用空格分开的两个整数，第一个是 n, 第二个是 m ( 0 < m,n <=300)。最后一行是：

0 0

输出

对于每行输入数据（最后一行除外)，输出数据也是一行，即最后猴王的编号

样例输入

6 2

12 4

8 3

0 0

样例输出

5

1

7

请完全按照如下的程序书写代码，并在书写的过程中体会优秀的代码风格：

#include<iostream>

using namespace std;

//一共最多有300只猴子

int succedent[300]; //这个数组用于保存一个猴子后一位是谁，

//比如“next[5]的值是7”就是说5号猴子的下一位是7号猴子，6号猴子已经在之前退出了。

int precedent[300];//这个数组用于保存一个猴子前一位是谁，用法和上面的类似。

int main() {

int n, m;

while (true) {

cin >> n >> m;

if (n == 0 && m == 0)

break;

for (int i = 0; i < n - 1; i++) {

succedent[i] = i + 1;

precedent[i + 1] = i;

}

succedent[n - 1] = 0;

precedent[0] = n - 1;

int current = 0;

while (true) {

//如果一共要报m次号，那么取m-1次succedent之后就是需要退出的那只猴子

for (int count = 0; count < m-1; count++)

current = succedent[current];

int pre = precedent[current];

int suc = succedent[current];

//让current号猴子退出很简单，就是把前一位的“下一位”指向current的下一位，

//下一位的“前一位”指向current的前一位就好了

succedent[pre] = suc;

precedent[suc] = pre;

if (pre == suc) {

//如果只剩下两个了，那么每个人的前位和后位就是同一个了。

//current是退出的，那么另一个就是剩下的。

//我们的序号是从0编号的，输出时要加一

cout << pre+1 << endl;

break;

}

current = suc;

}

}

return 0;

}

抄写题＃2：分数求和

[来源: POJ](http://pkuic.openjudge.cn/practice/1006/) (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

**注意： 总时间限制: 1000ms 内存限制: 65536kB**

描述

输入n个分数并对他们求和，用约分之后的最简形式表示。

比如：

q/p = x1/y1 + x2/y2 +....+ xn/yn，

q/p要求是归约之后的形式。

如：5/6已经是最简形式，3/6需要规约为1/2, 3/1需要规约成3，10/3就是最简形式。

PS:分子和分母都没有为0的情况，也没有出现负数的情况

输入

第一行的输入n,代表一共有几个分数需要求和

接下来的n行是分数

输出

输出只有一行，即归约后的结果

样例输入

2

1/2

1/3

样例输出

5/6

请完全按照如下的程序书写代码，并在书写的过程中体会优秀的代码风格：

#include<iostream>

using namespace std;

int main() {

int n;

cin >> n;

int sumn = 0, sumd = 1;//储存结果，sumn/sumd

while (n--) {

int num, deno;

char slash;//专门用来吃掉/的

cin >> num >> slash >> deno;

//先相加 a/b + c/d = (a\*d+c\*b)/(b\*d)

sumn = sumn\*deno + num\*sumd;

sumd = sumd\*deno;

}

//后约分

//先求最大公约数gcd，这里用的是欧几里得法

int a = sumd, b = sumn, c;

while (a != 0) {

c = a; a = b%a; b = c;

}

int gcd = b;

//分子分母同时除以gcd就可以完成约分

sumd /= gcd;

sumn /= gcd;

if (sumd > 1)

cout << sumn << '/' << sumd << endl;

else

cout << sumn << endl;

return 0;

}

//我们计算过程中结果分母是不断乘以新输入的分母，最后约分的。这样可能导致这个过程中分母过大溢出。

//这道题的数据比较简单，并没有出现那种情况。但大家可以思考一下，如果出现了那种情况怎么办呢？（不要用大整数啊）

/\*我给大家一组测试数据，看看你修改过的程序能不能通过这组数据吧：

样例输入：

2

1/100000000

1/100000000

样例输出：

1/50000000

\*/