2007年3月26日，在中俄两国元首的见证下，中国国家航天局局长孙来燕与俄罗斯联邦航天局局长别尔米诺夫共同签署了《中国国家航天局和俄罗斯联邦航天局关于联合探测火星-火卫一合作的协议》，确定中俄双方将于2009年联合对火星及其卫星“火卫一”进行探测。   
  
而卫星是进行这些探测的重要工具，我们的问题是已知两颗卫星的运行周期，求它们的相遇周期。

**Input**

输入数据的第一行为一个正整数T, 表示测试数据的组数. 然后是T组测试数据. 每组测试数据包含两组正整数，用空格隔开。每组包含两个正整数，表示转n圈需要的天数(26501/6335，表示转26501圈要6335天)，用'/'隔开。

**Output**

对于每组测试数据, 输出它们的相遇周期，如果相遇周期是整数则用整数表示，否则用最简分数表示。

**Sample Input**

2

26501/6335 18468/42

29359/11479 15725/19170

**Sample Output**

81570078/7

5431415

其实就是求 a / b, c / d 的最小公倍数x / y

 对于两个最简的分数 a / b, c / d 把他们两个的最小公倍数 x / y 也设为一个分数形式，那么这个 x 一定能够被 a , c整除， y 一定能够整除 b , d。那么要求得最小公倍数，那么肯定是分子尽量小，即 a , c 的最小公倍数， 分母尽量大， 即 b , d 的最大公约数。

如果a,b互质，cd互质，那么a\*c与c\*d也是互质

如果bd互质，那么他们的公共周期就只能是两个的最小公倍数了，时间没法约了

#include <iostream>

#include<stdio.h>

using namespace std;

int gcd(int a,int b)

{

if(b==0)return a;

else return gcd(b,a%b);

}

int lcm(int a,int b)

{

return a/gcd(a,b)\*b;

}

int main()

{

// freopen("input.txt","r",stdin);

int n,t,a,b,c,d;

cin>>n;

while(n--)

{

scanf("%d/%d %d/%d",&a,&b,&c,&d);

t=gcd(a,b);

a/=t;b/=t;

t=gcd(c,d);

c/=t;d/=t;

if(gcd(b,d)==1)cout<<lcm(a,c)<<"\n";

else

cout<<lcm(a,c)<<'/'<<gcd(b,d)<<"\n";

}

return 0;

}

注意这种带字符的读入方式