**算法训练 Hankson的趣味题**

时间限制：1.0s   内存限制：64.0MB

问题描述

　　Hanks 博士是BT (Bio-Tech，生物技术) 领域的知名专家，他的儿子名叫Hankson。现 在，刚刚放学回家的Hankson 正在思考一个有趣的问题。 今天在课堂上，老师讲解了如何求两个正整数c1 和c2 的最大公约数和最小公倍数。现 在Hankson 认为自己已经熟练地掌握了这些知识，他开始思考一个“求公约数”和“求公 倍数”之类问题的“逆问题”，这个问题是这样的：已知正整数a0,a1,b0,b1，设某未知正整 数x 满足： 1． x 和a0 的最大公约数是a1； 2． x 和b0 的最小公倍数是b1。 Hankson 的“逆问题”就是求出满足条件的正整数x。但稍加思索之后，他发现这样的 x 并不唯一，甚至可能不存在。因此他转而开始考虑如何求解满足条件的x 的个数。请你帮 助他编程求解这个问题。

输入格式

　　输入第一行为一个正整数n，表示有n 组输入数据。  
  
　　接下来的n 行每 行一组输入数据，为四个正整数a0，a1，b0，b1，每两个整数之间用一个空格隔开。输入 数据保证a0 能被a1 整除，b1 能被b0 整除。

输出格式

　　输出共n 行。每组输入数据的输出结果占一行，为一个整数。  
　　对于每组数据：若不存在这样的 x，请输出0； 若存在这样的 x，请输出满足条件的x 的个数；

样例输入

2  
41 1 96 288  
95 1 37 1776

样例输出

6  
2

样例说明

　　第一组输入数据，x 可以是9、18、36、72、144、288，共有6 个。  
　　第二组输入数据，x 可以是48、1776，共有2 个。

数据规模和约定

　　对于 50%的数据，保证有1≤a0，a1，b0，b1≤10000 且n≤100。  
　　对于 100%的数据，保证有1≤a0，a1，b0，b1≤2,000,000,000 且n≤2000。

锦囊1

枚举或数论方法。

锦囊2

x是a1的倍数，b1的约数，可以枚举b1所有的约数来判断是否满足条件。 也可以使用数论的方法，将a0, a1, b0, b1分解因数，可以找到x对于每个质因子的范围，根据这个可以得到答案的公式（将每个质因子的范围相乘）。

本题的C++参考代码如下：

#include<stdio.h>

#include<math.h>

const int maxn = 55000;

const int pmaxn = 5500;

int prime[pmaxn], p, ans = 1, tt;

bool vis[maxn];

int div(int & a, int b)

{

int cnt = 0;

while(a % b == 0)

{

a /= b;

cnt++;

}

return cnt;

}

int main()

{

int test = 50010;

int t = 0, n, m = (int)sqrt(maxn+0.5), i;

for(i = 2; i <= m; i++)

{

if(vis[i]) continue;

prime[t++] = i;

for(int j = i \* i; j < maxn; j += i)

vis[j] = true;

}

for(; i < maxn; i++)

if(!vis[i])

prime[t++] = i;

scanf("%d", &n);

while(n--)

{

int a, aa, b, bb;

ans = 1;

tt = 0;

scanf("%d%d%d%d", &a, &aa, &b, &bb);

for(i = 0; bb != 1; i++)

{

if(i > 4950)

{

prime[i] = bb;

// break;

}

if(bb % prime[i] == 0)

{

int cnt = 100000000, type;

int a1c = div(aa, prime[i]);

int a0c = div(a, prime[i]);

if(a1c > a0c) continue;

if(a1c == a0c)

{

cnt = a1c;

type = 1;

}

if(a1c < a0c)

{

type = 2;

cnt = a1c;

}

int b0c = div(b, prime[i]);

int b1c = div(bb, prime[i]);

int tot;

if(b1c == b0c)

{

if(type == 1)

tot = b1c - cnt + 1;

else

{

if(cnt > b1c)

continue;

tot = 1;

}

if(tot <= 0) continue;

}

if(b1c > b0c)

{

if(type == 2)

{

if(b1c != cnt)

{

ans = 0;break;

}

tot = 1;

}

else

{

if(cnt > b1c)

continue;

tot = 1;

}

}

ans \*= tot;

}

}

printf("%d\n", ans);

}

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

int max(int x,int y)

{

while(y>0)

{

int t=x%y;

x=y;

y=t;

}

return x;

}

int main()

{

int i;

int t1,t2,t3,t;

int n;

int cnt;

int a0,a1,b0,b1;

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d %d %d %d",&a0,&a1,&b0,&b1);

if(b1%a1!=0)

{

printf("0\n");

continue;

}

cnt=0;

t1=a0/a1;

t2=b1/b0;

t3=b1/a1;

for(t=1;t\*t<t3;t++)

if(t3%t==0)

{

if(max(t1,t)==1&&max(t2,t3/t)==1)cnt++;

if(max(t1,t3/t)==1&&max(t2,t)==1)cnt++;

}

if(t\*t==t3&&(max(t1,t)==1&&max(t2,t)==1))cnt++;

printf("%d\n",cnt);

}

return 0;

}

本题的Java参考代码如下：

import java.util.Scanner;

import java.util.ArrayList;

public class Main{

private static ArrayList<Integer> list = new ArrayList<Integer>();

private static Scanner sc = new Scanner(System.in);

private static int[] p = new int[20000];

private static int[] a = new int[200000];

private static int[] num = new int[20000];

private static int end;

private static int len;

public static void main(String[] args)

{

int n = sc.nextInt();

while (n > 0)

{

int a0 = sc.nextInt();

int a1 = sc.nextInt();

int b0 = sc.nextInt();

int b1 = sc.nextInt();

int t = len = end = 0;

work(b1, a0, a1);

for (int i = 0; i < len; i++)

if ( gcd(a[i], a0) == a1 && b0 / gcd(b0, a[i]) \* a[i] == b1)

t++;

list.add(t);

n--;

}

for (int i : list)

System.out.println(i);

}

public static int gcd(int a, int b)

{

int t;

while (b != 0)

{

t = a % b;

a = b;

b = t;

}

return a;

}

public static void dfs(int now, int pro)

{

if (now == end)

{

a[len++] = pro;

return;

}

dfs(now + 1, pro);

for (int i = 1; i <= num[now]; i++)

{

pro \*= p[now];

dfs(now + 1, pro);

}

}

public static void work(int n, int a0, int a1)

{

int i = 2;

int lim = (int) Math.sqrt(n);

while (n != 1 && i <= lim)

{

if (n % i == 0)

{

if (a1/i == 0 && a1%i == 0)

{

while (n % i == 0)

n /= i;

i++;

continue;

}

p[end] = i;

num[end] = 0;

while (n % i == 0) {

num[end]++;

n /= i;

}

end++;

}

i++;

}

if (n != 1) {

p[end] = n;

num[end++] = 1;

}

dfs(0, 1);

}

}