**Sachin and Varun**

Varun and you are talking about a movie that you have recently watched while Sachin is busy teaching Number Theory. Sadly, Sachin caught Varun talking to you and asked him to answer his question in order to save himself from punishment. The question says:

Given two weights of a and b units, in how many different ways you can achieve a weight of d units using only the given weights? Any of the given weights can be used any number of times (including 0 number of times).

Since Varun is not able to answer the question, he asked you to help him otherwise he will get punished.

Note: Two ways are considered different if either the number of times a is used or number of times b is used is different in the two ways.

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

class Triplet{

public:

ll gcd;

ll x;

ll y;

};

Triplet gcdExtendedEuclid(ll a,ll b){

if(b==0){

Triplet myAns;

myAns.gcd = a;

myAns.x = 1;

myAns.y = 0;

return myAns;

}

Triplet smallAns = gcdExtendedEuclid(b,a%b);

Triplet myAns;

myAns.gcd = smallAns.gcd;

myAns.x = smallAns.y;

myAns.y = (smallAns.x - ((a/b)\*(smallAns.y)));

return myAns;

}

ll modInverse(ll A, ll M)

{

ll x = gcdExtendedEuclid(A, M).x;

return (x % M + M) % M;

}

int main(){

int t;

cin>>t;

while(t--){

ll a,b,d;

cin >> a >> b >> d;

ll g = \_\_gcd(a,b);

if(d%g){

cout << 0 << endl;

continue;

}

if(d == 0){

cout << 1 << endl;

continue;

}

a/=g;

b/=g;

d/=g;

ll y1 = ((d%a) \* modInverse(b,a) ) %a;

ll firstValue = d/b;

if(d < y1\*b){

cout << 0 <<endl;

continue;

}

ll n = (firstValue - y1)/a;

ll ans = n +1 ;

cout << ans <<endl;

}

return 0;

}