Ichigo and Cubes

Let P,Q,R be the dimensions of the box with (P,Q) = (Q,R) = (R,P) = 1.
Assume that the slicing plane is given by the equation x/P + y/Q + z/R = 1 (it passes through (P,0,0), (0,Q,0) and (0,0,R)).
First, in the box bounded by 0<=x<=P, 0<=y<=Q and 0<=z<=R, this plane passes only through three lattice points: (P,0,0), (0,Q,0), (0,0,R).
Why? Because if x/P + y/Q + z/R = 1 with (x,y,z) a lattice point, then xQR = P(QR - yR - zQ).
Since P is coprime to Q and R, it follows that P divides x. similarly, Q divides y and R divides z, and therefore the only possibilities are (P,0,0), (0,Q,0), and (0,0,R).
Let (i,j,k) be a triple such that i/P + j/Q + k/R < 1, i,j,k>=0 and with one of i, j, k equal to 0.
Consider the sequence of points (i,j,k), (i+1,j+1,k+1), (i+2,j+2,k+2), ...
In this sequence, eventually we will find a point (I,J,K) such that I/P + J/Q + K/R > 1.
I claim that the cube whose corners are (I-1,J-1,K-1) and (I,J,K) is sliced by the plane.
This is because the plane never passes through a lattice point, and since it passes through the segment (I-1,J-1,K-1) and (I,J,K),
it therefore slices the cube with those corners.
Therefore, the plane passes through exactly one cube in this sequence of corners. (Exactly one because all other cubes has corners either strictly below or above the plane)
So the answer is the number of such sequences:
F(P,Q,R) = #{(i,j,k) : i/P + j/Q + k/R < 1 and i,j,k>=0 and one of i, j, k is equal to 0}
Let G(A,B) be the number of integer pairs (i,j) such that i/A + j/B < 1 and i,j >= 0. Then with inclusion-exclusion, we have
F(P,Q,R) = G(P,Q) + G(Q,R) + G(R,P) - P - Q - R + 1
Now consider G(A,B) (with (A,B) = 1). We have:
G(A,B)
= #{(i,j) : i/A + j/B < 1 and i,j >= 0}
= #{(i,j) : j < B - i*B/A and i,j >= 0}
= sum(0<=i = sum(0<=i = sum(0<=i = AB - sum(0<=i = AB - B/A * sum(0<=i Now, the first sum is simply A(A-1)/2. The second sum is a bunch of integers scattered in the range [0,A). However, since (A,B) = 1, we see that it actually spans the whole range [0,A) ! Therefore it is also A(A-1)/2.
= AB - B/A * (A(A-1)/2) - 1/A (A(A-1)/2)
= (2AB-(AB-A-B+1))/2
= (AB+A+B-1)/2
Therefore:
F(P,Q,R)
= G(P,Q) + G(Q,R) + G(R,P) - P - Q - R + 1
= ((PQ+P+Q-1) + (QR+Q+R-1) + (RP+R+P-1))/2 - P - Q - R + 1
= (PQ+QR+RP-1)/2
= floor((PQ+QR+RP)/2)
where the last one follows because PQ+QR+RP is always odd (follows from the fact that P,Q,R are pairwise coprime.)

Statistics
Difficulty: Medium - Ha
Success Rate: 94.87%
Time Complexity: O(1)
Required Knowledge: Bi
Inclusion Exclusion
Publish Date: Apr 29 20

Originally featured in A
Math Programming Con
Of the 1581 contest pa
(2.47%) submitted co
challenge.

Problem Setter's code :

```
#include <bits/stdc++.h>

using namespace std;
struct _ { ios_base::Init i; _() { cin.sync_with_stdio(0); cin.tie(0); } } _;

const long long mod = 1000000007;

long long gcd(long long a, long long b)
```

```
{
    if(a < 0) a = -a; if(b < 0) b = -b;
    return b ? gcd(b, a % b) : a;
}

int main()
{
    int t; cin >> t;
    while(t--)
    {
        long long p, q, r;
        cin >> p >> q >> r;
        assert(gcd(p, q) == 1);
        assert(gcd(p, r) == 1);
        assert(gcd(q, r) == 1);
        p %= mod, q %= mod, r %= mod;

        long long ret = (p * q) % mod;
        ret += (q * r) % mod;
        if(ret >= mod) ret -= mod;
        ret += (r * p) % mod;
        if(ret >= mod) ret -= mod;

        if(--ret < 0) ret += mod;
        if(ret & 1) ret += mod;

        cout << (ret >> 1) << '\n';
    }
    return 0;
}
```

Problem Tester's code :

```
for z in xrange(input()): print (lambda P,Q,R: P*Q+Q*R+R*P>>1)(*map(int, raw_input().str
ip().split()))%(10**9+7)
```