Name → Gaurav Singh
UID → 23BCS12992
Sec → KRG 1-A

Q1 → Given the three integers n, a and b, find the n^{th} magical number.
A number X is magical if it is either divisible by a or b.
If answer is very large, return it modulo 10^9+7.

n=1, a=2, b=3
ans → 2

n=4, a=2, b=3
ans → 6

$n \approx 10^9$

Approach →

Binary search →

lower limit of binary search → min (a, b)

max^m limit of binary search → n × min (a, b).

If we want to find the number of multiples of either a or b, from 1 to X, we can calculate it through →

Number of multiples of a or b till X → $\frac{X}{a} + \frac{X}{b} - \frac{X}{lcm(a,b)}$

Calculate number of multiples at every mid and compare with n.
→ Apply binary search accordingly.

```
using namespace std;
#define ll long long
# define mod 100000007
int solve (int n, int a, int b) {
    ll low = min(a,b);
    ll high = 1LL * n * min(a,b);
    ll lcm = 1LL * (a*b)/__gcd(a,b);
    while (low < high) {
        ll mid = low + (high - low)/2;
        ll cnt = mid/a + mid/b - mid/lcm;
        if (cnt < n) { low = mid+1;
        else high = mid;
    }
    return (low % mod);
}
int main() {
    int n,a,b;
    cin>>n>>a>>b;
    cout<< solve(n,a,b) << endl;
}
```