-Amrit Singh -23BCS12931 246 $^{n-1}C_r + ^{n}C_{r-1}$ $^{n-1}C_{r-1} + ^{n-1}C_r$

Q-1) Given the three integers n, a, b, return the nth magical number. A positive integer is magical number if it is divisible by either a or b

$1 \leq n \leq 10^9$

$1 \leq a \leq 10^5$

→ **Approach:**

→ Brute Force: $O(n)$ We start from 1 and then check if it is divisible by a or b and have a count to count frequency.

→ Not optimal for large range.

→ Efficient Approach

$O(\log n)$ (Binary Search) + (Inclusion - exclusion)

→ $l = \min(a, b)$

→ $r = n * \min(a, b)$

→ First find a mid then check count by:

$$\left[\frac{mid}{a}\right] + \left[\frac{mid}{b}\right] + \left|\frac{mid}{lcm\,a,b}\right|$$

→ If count < n then move to right

else if count >= n move to left.

→ Code:
```
int Magical (int n, int a, int b) {
~~long~~ long l = Min(a, b);
long long r = n * min(a, b);
long long ans;
while (l <= r) {
long long mid = l + (r-1)/2 ;
long long count = ~~Floor~~ mid/a + mid/b ~~*~~ - mid/lcm(a, b)
if (count < n) {
    l = mid + 1; }
else if (count >= n) {
    ans = mid;
    mid = r - 1; }
  }
return ans;
```

```
int GCD (int a, int b){
        if (b==0) return a
    return (b ,a % b);}


int lcm (int a, int b){
        return 1LL* a* b/GCD(a,b);}
```