Given

n, a, b return n^{th} magical no. A positive integer is magical when it is divisible by a or b

n = 1, a = 2, b = 3
o/p = 2

n = 4, a = 2, b = 3
o/p = 6 (2 3 4 6)

```
count = 0
num = min(a, b)                    # Brute Force
while (count != n){
    if (num % a == 0 || num % b == 0) {
        count ++;
    } num ++;
}
return num;
```

---

well due to constraints $1 \le n \le 10^9$
num can overflow, TLE hit
So we will use different Approach

a = 2, multiples of 2 : 2, 4, 6, 8, ...
b = 3, multiples of 3 : 3, 6, 9, 12, ...

(mul of 2) ∪ (mul of 3) = {2, 3, 4, 6, 8, 9, 10, 12, ...}

So we will look into this series only

App ⇒

```
if (num % a == 0){
    count ++;
    num += a;
}
if (num % b == 0) {
    count ++;
    num += b;
}
```

WHAT About LCM?

⇒

```
if (num % a == 0 && num % b == 0){
    count --;
}
```

This approach greatly optimises the algor