Name → Anmalpreet Singh   UID → 23BCS11616
Class → KRG-1A!

Q.→ Given the three integers n, a and b
return the nth magical number. Since
the answer is very large return $10^9 +$;
A positive integer is magical no. when
number is divisible by either a or
b.

Test case ÷ ① n=1, a=2, b=3
                outcome = 2

② n=4, a=2, b=3
   outcome = 6

Approach → ① Counting magical numbers $\leq x$

② Binary Search on answer

③ Search boundaries
   left = 1
   Right = n * min(a,b)

④ Binary search steps

⑤ Modulo

```
Code :- Class solution:
          def nthMagical(se n, int a, int b);
          mod)= 10**9+7
          def gcd(x, y):
              while y
                  x, y = y, x%y
              return x
          lcm = a*b // gcd(a,b)
          left, right= 1, n* min(a,b)
          while left < right:
              mid = (left +right) //2
              count = mid // a +mid//b - mid// lc
              if count <n;
                  left = m+1
          else:
                  right = mid
          return left%. mod

print(Solution(). nthMagical(5,2,3))
```

Output :- 8

Time Complexity → $O(\log(n * \min(a,b))$

Space Complexity → $O(1)$

```python
class Solution:
    def nthMagicalNumber(self, n: int, a: int, b: int) -> int:
        MOD = 10**9 + 7

        def gcd(x, y):
            while y:
                x, y = y, x % y
            return x

        lcm = a * b // gcd(a, b)

        left, right = 1, n * min(a, b)

        while left < right:
            mid = (left + right) // 2
            count = mid // a + mid // b - mid // lcm

            if count < n:
                left = mid + 1
            else:
                right = mid

        return left % MOD


print(Solution().nthMagicalNumber(5, 2, 3))
```