

Name : Aaditya Anand
Ud : 23BCS10446
Sec : KRG-3A

30/01/2026

Question:

Given three integers n, a and b, return n^{th} magical no. Since the Ans may be very large return mod $10^9 + 7$

A magical no - if no. is divisible by a or b

$n=1$

a=2

Output = 2

b=3

Code:

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

```
using namespace std;
```

```
class Solution {
```

```
public:
```

```
    static const int MOD = 1000000007;
```

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

```
        if (b == 0) {
```

```
            return a; }
```

```
        return gcd(b, a % b); }
```

```
    long long lcm (long long a, long long b) {
```

```
        long long Ans = (a / gcd(a,b)) * b;
```

```
        return Ans; }
```

```
int nthMagicalNumber (int n, int a, int b) {
```

```
    long long low = min(a,b);
```

```
    long long high = (long long)n * low ;
```

```
    long long L = LCM(a,b);
```

```
    while (low < high) {
```

```
        long long mid = low + (high - low)/2;
```

```

long long Count = mid/a + mid/b - mid/L;
if (Count < n) {
    low = mid + 1;
} else {
    high = mid;
}

```

```
return low % MOD; } }
```

```
int main() {
```

```
    Solution obj;
```

```
    int n, a, b;
```

```
    cout << "Enter n, a, b: ";
```

```
    cin >> n;
```

```
    cin >> a;
```

```
    cin >> b;
```

```
    int result = obj.nthMagicalNumber(n, a, b);
```

```
    cout << "Nth magical number = " << result <<
```

```
    endl;
```

```
    return 0; }
```

Algorithm:

1. Start
2. Create a function $\text{gcd}(a, b)$ to find Greatest Common Divisor.
3. If $b = 0$, return a .
4. Else, return $\text{gcd}(b, a \bmod b)$
5. Create a function $\text{lcm}(a, b)$ to find Least Common Multiple.
6. $\text{lcm} = (a \times b) / \text{gcd}(a, b)$
7. Return LCM
8. Create a function $\text{nthMagicalNumber}(n, a, b)$.
9. $\text{low} = \min(a, b)$ and $\text{high} = n \times \min(a, b)$
10. Find $L = \text{lcm}(a, b)$.

- while low < high, repeat :
- mid = low + (high - low) / 2
- Count = (mid/a) + (mid/b) - (mid/L)
- if Count < n:
- low = mid + 1
- else:
- high = mid
- when low = high, Store low as answer.
- return answer = low mod(10⁹ + 7)
- Read values n, a, b.
- Call nth magical Number (n, a, b).
- Display the result.
- END.

Output

$$n = 5$$

$$a = 2$$

$$b = 3$$

$$\text{nth magical Number} = 8$$