

Ques! Given three Integers n, a and b . A number is magical no. they are magical if it is divisible by either a or b . Since, the answer can be large use 10^9+7 .

eg $\Rightarrow n=1, a=3, b=2$
 $OP \Rightarrow 2$

ALGORITHM \Rightarrow

- 1) Compute the GCD
 \Rightarrow find GCD using Euclid's Algorithm
- 2) Compute The LCM
 \Rightarrow Since $a \times b = LCM \times GCD \Rightarrow LCM = (a/GCD) \times b$
- 3) Define Search space.
 $low = \min(a, b);$
 $High = n \times \min(a, b);$ because $n \times (\min(a, b))$ will for sure be n^{th} special No.
- 4) Binary Search

```

while (l <= r) {
    mid = l + (r - l) / 2;
    Compute (mid/a + mid/b - mid/LCM)
    if (count > n) {
        r = mid - 1;
    } else {
        l = mid;
    }
}

```

T.C. $\Rightarrow \log n$
 S.C. $\Rightarrow O(1)$

5) Print ans as the n^{th} special number if we get $\text{Count} == n$.

```
int rthSpecial (int a, int b, int c) {
```

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

```
    int l = min(a, b);
```

```
    int r = n * min(a, b);
```

```
    int ans = -1;
```

```
    while (l <= r) {
```

```
        int mid = l + (r - l) / 2;
```

```
        if (Valid (mid, a, b, LCM, n)) {
```

```
            ans = mid;
```

```
            r = mid - 1;
```

```
        else
```

```
            l = mid + 1;
```

```
    return ans;
```

```
bool Valid (int mid, int a, int b, int LCM, int n) {
    long Count = mid / a + mid / b - mid / LCM;
    return Count >= n;
}
```

```
int GCD (int a, int b) {
```

```
    return b == 0 ? a : GCD(b, a % b);
}
```