

Ques 1

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

eg) $n = 1 \quad d = 3 \quad a, b = 2$

$$\text{OP} \Rightarrow 2$$

$$d * ((a \cdot a) \bmod a) = \text{mod } a$$

ALGORITHM :-

1) Compute the GCD

\Rightarrow find GCD using Euclid's algorithm

2) Compute the LCM

$$\Rightarrow \text{Since } a \times b = \text{LCM} \times \text{GCD} \Rightarrow \text{LCM} = (a/\text{GCD}) \times b$$

3) Define Search space.

$$\text{low} = \min(a, b); \quad \text{high} = \max(a, b)$$

High - $n \times \min(a, b)$; because $n \times (\min(a, b))$ will for sure be n^{th} Special No.

4) Binary Search

while ($l \leq r$) {

$$\text{mid} = l + (r - l)/2;$$

Compute $\left(\frac{\text{mid}}{a} + \frac{\text{mid}}{b} - \frac{\text{mid}}{\text{LCM}} \right)$

T.C. $\Rightarrow \log n$

S.C. $\Rightarrow O(1)$

Beta performs Binary Search

if $(\text{count} > n \cdot \{ \text{hi} \})$ then

$$r = \text{mid} - 1$$

else

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

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

$$\text{long LCM} = (a / \text{gcd}(a, b)) * b;$$

```
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;
```

```
boolean Valid (int mid, int a, int b, int LCM, int n) {
```

$$\text{long Count} = \frac{\text{mid}}{a} + \frac{\text{mid}}{b} - \frac{\text{mid}}{\text{LCM}}$$

```
return Count >= n;
```

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

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