

The screenshot shows a programming challenge titled "Multiply by 2, divide by 6". It includes a Java code snippet, a problem statement, and a test case table.

```

public class Main {
    public static void main(String[] args) {
        int n = 10;
        System.out.println(minOperations(n));
    }

    public static int minOperations(int n) {
        if (n == 1) return 0;
        if (n % 6 == 0) return minOperations(n / 6);
        else return minOperations(n * 2) + 1;
    }
}

```

B. Multiply by 2, divide by 6

Given an integer n , find the minimum number of operations needed to obtain 1 from n . You can either multiply by 2 or divide by 6 (if it is divisible by 6 without the remainder).

For example, if $n = 10$, the answer is 3 because we can do the following:

$$\begin{aligned} & 10 \rightarrow 20 \\ & 20 \rightarrow 10 \\ & 10 \rightarrow 1 \end{aligned}$$

or

$$\begin{aligned} & 10 \rightarrow 6 \\ & 6 \rightarrow 3 \\ & 3 \rightarrow 1 \end{aligned}$$

or

$$\begin{aligned} & 10 \rightarrow 12 \\ & 12 \rightarrow 6 \\ & 6 \rightarrow 3 \\ & 3 \rightarrow 1 \end{aligned}$$

or

$$\begin{aligned} & 10 \rightarrow 20 \\ & 20 \rightarrow 12 \\ & 12 \rightarrow 6 \\ & 6 \rightarrow 3 \\ & 3 \rightarrow 1 \end{aligned}$$

or

$$\begin{aligned} & 10 \rightarrow 20 \\ & 20 \rightarrow 40 \\ & 40 \rightarrow 20 \\ & 20 \rightarrow 10 \\ & 10 \rightarrow 1 \end{aligned}$$

Test Case	Input (n)	Output (minOperations(n))
1	1	0
2	2	1
3	3	1
4	4	2
5	5	2
6	6	1
7	7	3
8	8	2
9	9	2
10	10	3

give prove of this

Here is **the complete proof** for the solution logic of CF 1374B — Multiply by 2, divide by 6.

✓ Goal

You are given a number n .

You can do two operations:

1. **Multiply by 2** → $n = n * 2$
2. **Divide by 6** (only if $n \% 6 == 0$) → $n = n / 6$

You need the **minimum steps** to reach 1 from n .

✓ Key Insight

To reach 1 , the prime factorization must end up as:

$$1 = 2^0 \cdot 3^0.$$

So n must contain only primes 2 and 3.

If any other prime appears → **impossible (return -1)**.

✓ STEP 1 — Factorize

Let:

- cnt2 = number of times 2 divides n
- cnt3 = number of times 3 divides n

Example:

$$n = 72 = 2^3 * 3^2$$

→ `cnt2 = 3, cnt3 = 2`

If after dividing out all 2's and 3's, the leftover is not 1 →
there was some other prime → **impossible**.

✓ STEP 2 — Understanding operations in terms of prime factors

Operation 1: multiply by 2

$$n \leftarrow n \times 2$$

→ increases exponent of 2:

$$2^{cnt2} \cdot 3^{cnt3} \rightarrow 2^{cnt2+1} \cdot 3^{cnt3}$$

Operation 2: divide by 6

$$n \leftarrow \frac{n}{6} = \frac{n}{2 \cdot 3}$$

→ reduces exponent of 2 and 3:

$$2^{cnt2} \cdot 3^{cnt3} \rightarrow 2^{cnt2-1} \cdot 3^{cnt3-1}$$

✓ STEP 3 — What must happen to reach 1?

We want final:

$$2^0 \cdot 3^0$$

After divisions, both exponents must reach zero.

Every **divide by 6** decreases:

- exponent of 2 → minus 1
- exponent of 3 → minus 1

This is the only operation that reduces 3's.

Critical Observation

To remove `cnt3` threes, we need exactly `cnt3` divisions by 6.

Each division also removes one 2.

So after removing all 3's:

$$\text{Remaining 2's} = \text{cnt2} - \text{cnt3}$$

To be valid, after all operations we must reach exponent of 2 equal to 0.

Thus:

$$\text{cnt2} - \text{cnt3} \leq 0$$

→ If:

✗ `cnt2 > cnt3` → impossible

Because divide-by-6 removes 2's and 3's equally.

You don't have enough 3's to "pair" with extra 2's.

✓ STEP 4 — If valid: compute moves

We need:

- exactly `cnt3` divide-by-6 operations (to kill 3's)
- if `cnt2 < cnt3`, we must **multiply by 2** to increase exponent of 2 until they match

Each multiply-by-2 increases exponent of 2 by 1.

We need:

$$(\text{cnt3} - \text{cnt2}) \text{ multiplications}$$

Total moves:

$$(\text{cnt3} - \text{cnt2}) + \text{cnt3} = 2 \cdot \text{cnt3} - \text{cnt2}$$

This matches code:

 Copy code

```
2 * cnt3 - cnt2
```

✓ FINAL FORMULA (Proven)

If n contains primes other than 2 or 3 → -1

Else:

- If `cnt2 > cnt3` → -1
- Else:

$$\text{answer} = (cnt3 - cnt2) + cnt3 = 2cnt3 - cnt2$$

🔥 Example Proof Works

Example: `n = 12 = 2^2 * 3^1`

`cnt2 = 2, cnt3 = 1`

Check: `cnt2 > cnt3`

→ impossible

Output: -1

Matches sample.

Example: `n = 36 = 2^2 * 3^2`

`cnt2 = 2, cnt3 = 2`

valid

Moves:

- need `cnt3 = 2` divides
- need `cnt3 - cnt2 = 0` multiplies

Total = `2*2 - 2 = 2`

Matches.

🎉 Final Result

This proves the CF editorial logic:

✓ If other primes → return -1

✓ If $\text{cnt2} > \text{cnt3}$ → return -1

✓ Else return $2 \cdot \text{cnt3} - \text{cnt2}$

If you want, I can also give you a **step-by-step intuitive explanation**, a **diagram proof**, or help you **implement your own solution**.

