

# Ugly Number Problem ■

## ■ Problem Statement

An **ugly number** is a positive integer whose prime factors are limited to **2, 3, and 5**. You are given an integer **n**. Return **true** if **n** is an ugly number, otherwise return **false**.

## ■ Examples

**Input:** Input: n = 6

**Output:** Output: true

**Explanation:**  $6 = 2 \times 3 \rightarrow$  factors are 2 and 3 (valid ugly number ■).

**Input:** Input: n = 1

**Output:** Output: true

**Explanation:** 1 has no prime factors  $\rightarrow$  considered ugly by definition ■.

**Input:** Input: n = 14

**Output:** Output: false

**Explanation:**  $14 = 2 \times 7 \rightarrow$  contains prime factor 7 (not allowed ■).

## ■■ Constraints

$-2^{31} \leq n \leq 2^{31} - 1$

## ■ Approach

1. If  $n \leq 0$ , return false (since ugly numbers must be positive).
2. Continuously divide n by 2, 3, and 5 until it is no longer divisible.
3. If the final result is 1, then n is an ugly number; otherwise, it's not.

## ■■ Code Implementation (C++)

```
class Solution {
public:
    bool isUgly(int n) {
        if (n <= 0) return false;
        while (n % 2 == 0) n /= 2;
        while (n % 3 == 0) n /= 3;
        while (n % 5 == 0) n /= 5;
        return n == 1;
    }
};
```

## ■■ Time & Space Complexity

- **Time Complexity:**  $O(\log n)$

Because we are dividing by prime factors repeatedly.

- **Space Complexity:**  $O(1)$

No extra memory is used.