Problem 29: Divide Two Integers

Problem Information

Difficulty: Medium
Acceptance Rate: 19.03%
Paid Only: No
Tags: Math, Bit Manipulation

Problem Description

Given two integers `dividend` and `divisor`, divide two integers **without** using multiplication, division, and mod operator.

The integer division should truncate toward zero, which means losing its fractional part. For example, `8.345` would be truncated to `8`, and `-2.7335` would be truncated to `-2`.

Return _the**quotient** after dividing _`dividend` _by_`divisor`.

Note: Assume we are dealing with an environment that could only store integers within the **32-bit** signed integer range: `[–231, 231 – 1]`. For this problem, if the quotient is **strictly greater than** `231 - 1`, then return `231 - 1`, and if the quotient is **strictly less than** `-231`, then return `-231`.

Example 1:

Input: dividend = 10, divisor = 3 **Output:** 3 **Explanation:** 10/3 = 3.33333.. which is truncated to 3.

Example 2:

Input: dividend = 7, divisor = -3 **Output:** -2 **Explanation:** 7/-3 = -2.33333.. which is truncated to -2.

Constraints:

* `-231 <= dividend, divisor <= 231 - 1` * `divisor != 0`

Code Snippets

C++:

```
class Solution {
public:
int divide(int dividend, int divisor) {

}
};
```

Java:

```
class Solution {
public int divide(int dividend, int divisor) {

}
}
```

Python3:

```
class Solution:
def divide(self, dividend: int, divisor: int) -> int:
```