

Problem 1006: Clumsy Factorial

Problem Information

Difficulty: Medium

Acceptance Rate: 60.28%

Paid Only: No

Tags: Math, Stack, Simulation

Problem Description

The **factorial** of a positive integer n is the product of all positive integers less than or equal to n .

* For example, $\text{factorial}(10) = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1$.

We make a **clumsy factorial** using the integers in decreasing order by swapping out the multiply operations for a fixed rotation of operations with multiply $*$, divide $/$, add $+$, and subtract $-$ in this order.

* For example, $\text{clumsy}(10) = 10 * 9 / 8 + 7 - 6 * 5 / 4 + 3 - 2 * 1$.

However, these operations are still applied using the usual order of operations of arithmetic. We do all multiplication and division steps before any addition or subtraction steps, and multiplication and division steps are processed left to right.

Additionally, the division that we use is floor division such that $10 * 9 / 8 = 90 / 8 = 11$.

Given an integer n , return the clumsy factorial of n .

Example 1:

Input: $n = 4$ **Output:** 7 **Explanation:** $7 = 4 * 3 / 2 + 1$

Example 2:

Input: $n = 10$ **Output:** 12 **Explanation:** $12 = 10 * 9 / 8 + 7 - 6 * 5 / 4 + 3 - 2 * 1$

****Constraints:****

***`1 <= n <= 104`**

Code Snippets

C++:

```
class Solution {  
public:  
    int clumsy(int n) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int clumsy(int n) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def clumsy(self, n: int) -> int:
```