Normally, the factorial of a positive integer n is the product of all positive integers less than or equal to n.  For example, factorial(10) = 10 \* 9 \* 8 \* 7 \* 6 \* 5 \* 4 \* 3 \* 2 \* 1.

We instead make a *clumsy factorial:* using the integers in decreasing order, we swap out the multiply operations for a fixed rotation of operations: multiply (\*), divide (/), add (+) and subtract (-) in this order.

For example, 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 equals 11.  This guarantees the result is an integer.

Implement the clumsy function as defined above: given an integer N, it returns the clumsy factorial of N.

**Example 1:**

**Input:** 4

**Output:** 7

**Explanation:** 7 = 4 \* 3 / 2 + 1

**Example 2:**

**Input:** 10

**Output:** 12

**Explanation:** 12 = 10 \* 9 / 8 + 7 - 6 \* 5 / 4 + 3 - 2 \* 1

**Note:**

1. 1 <= N <= 10000
2. -2^31 <= answer <= 2^31 - 1  (The answer is guaranteed to fit within a 32-bit integer.)