You are standing at position 0 on an infinite number line. There is a destination at position target.

You can make some number of moves numMoves so that:

* On each move, you can either go left or right.
* During the ith move (starting from i == 1 to i == numMoves), you take i steps in the chosen direction.

Given the integer target, return *the****minimum****number of moves required (i.e., the minimum*numMoves*) to reach the destination*.

**Example 1:**

**Input:** target = 2

**Output:** 3

**Explanation:**

On the 1st move, we step from 0 to 1 (1 step).

On the 2nd move, we step from 1 to -1 (2 steps).

On the 3rd move, we step from -1 to 2 (3 steps).

**Example 2:**

**Input:** target = 3

**Output:** 2

**Explanation:**

On the 1st move, we step from 0 to 1 (1 step).

On the 2nd move, we step from 1 to 3 (2 steps).

**Constraints:**

* -109 <= target <= 109
* target != 0