**Stone Game IV:**

Alice and Bob take turns playing a game, with Alice starting first.

Initially, there are n stones in a pile.  On each player's turn, that player makes a *move* consisting of removing **any** non-zero **square number** of stones in the pile.

Also, if a player cannot make a move, he/she loses the game.

Given a positive integer n. Return True if and only if Alice wins the game otherwise return False, assuming both players play optimally.

**Example 1:**

**Input:** n = 1

**Output:** true

**Explanation:** Alice can remove 1 stone winning the game because Bob doesn't have any moves.

**Example 2:**

**Input:** n = 2

**Output:** false

**Explanation:** Alice can only remove 1 stone, after that Bob removes the last one winning the game (2 -> 1 -> 0).

**Example 3:**

**Input:** n = 4

**Output:** true

**Explanation:** n is already a perfect square, Alice can win with one move, removing 4 stones (4 -> 0).

**Example 4:**

**Input:** n = 7

**Output:** false

**Explanation:** Alice can't win the game if Bob plays optimally.

If Alice starts removing 4 stones, Bob will remove 1 stone then Alice should remove only 1 stone and finally Bob removes the last one (7 -> 3 -> 2 -> 1 -> 0).

If Alice starts removing 1 stone, Bob will remove 4 stones then Alice only can remove 1 stone and finally Bob removes the last one (7 -> 6 -> 2 -> 1 -> 0).

**Example 5:**

**Input:** n = 17

**Output:** false

**Explanation:** Alice can't win the game if Bob plays optimally.

**Constraints:**

* 1 <= n <= 10^5