You are given a **circular** array nums of positive and negative integers. If a number *k* at an index is positive, then move forward *k* steps. Conversely, if it's negative (-*k*), move backward *k* steps. Since the array is circular, you may assume that the last element's next element is the first element, and the first element's previous element is the last element.

Determine if there is a loop (or a cycle) in nums. A cycle must start and end at the same index and the cycle's length > 1. Furthermore, movements in a cycle must all follow a single direction. In other words, a cycle must not consist of both forward and backward movements.

**Example 1:**

**Input:** [2,-1,1,2,2]

**Output:** true

**Explanation:** There is a cycle, from index 0 -> 2 -> 3 -> 0. The cycle's length is 3.

**Example 2:**

**Input:** [-1,2]

**Output:** false

**Explanation:** The movement from index 1 -> 1 -> 1 ... is not a cycle, because the cycle's length is 1. By definition the cycle's length must be greater than 1.

**Example 3:**

**Input:** [-2,1,-1,-2,-2]

**Output:** false

**Explanation:** The movement from index 1 -> 2 -> 1 -> ... is not a cycle, because movement from index 1 -> 2 is a forward movement, but movement from index 2 -> 1 is a backward movement. All movements in a cycle must follow a single direction.

**Note:**

1. -1000 ≤ nums[i] ≤ 1000
2. nums[i] ≠ 0
3. 1 ≤ nums.length ≤ 5000