There are n rooms you need to visit, labeled from 0 to n - 1. Each day is labeled, starting from 0. You will go in and visit one room a day.

Initially on day 0, you visit room 0. The **order** you visit the rooms for the coming days is determined by the following **rules** and a given **0-indexed** array nextVisit of length n:

* Assuming that on a day, you visit room i,
* if you have been in room i an **odd** number of times (**including** the current visit), on the **next** day you will visit the room specified by nextVisit[i] where 0 <= nextVisit[i] <= i;
* if you have been in room i an **even** number of times (**including** the current visit), on the **next** day you will visit room (i + 1) mod n.

Return *the label of the****first****day where you have been in****all****the rooms*. It can be shown that such a day exists. Since the answer may be very large, return it **modulo** 109 + 7.

**Example 1:**

**Input:** nextVisit = [0,0]

**Output:** 2

**Explanation:**

- On day 0, you visit room 0. The total times you have been in room 0 is 1, which is odd.

  On the next day you will visit room nextVisit[0] = 0

- On day 1, you visit room 0, The total times you have been in room 0 is 2, which is even.

  On the next day you will visit room (0 + 1) mod 2 = 1

- On day 2, you visit room 1. This is the first day where you have been in all the rooms.

**Example 2:**

**Input:** nextVisit = [0,0,2]

**Output:** 6

**Explanation:**

Your room visiting order for each day is: [0,0,1,0,0,1,2,...].

Day 6 is the first day where you have been in all the rooms.

**Example 3:**

**Input:** nextVisit = [0,1,2,0]

**Output:** 6

**Explanation:**

Your room visiting order for each day is: [0,0,1,1,2,2,3,...].

Day 6 is the first day where you have been in all the rooms.

**Constraints:**

* n == nextVisit.length
* 2 <= n <= 105
* 0 <= nextVisit[i] <= i