Given an array of positive integers nums, remove the **smallest** subarray (possibly **empty**) such that the **sum** of the remaining elements is divisible by p. It is **not** allowed to remove the whole array.

Return *the length of the smallest subarray that you need to remove, or*-1*if it's impossible*.

A **subarray** is defined as a contiguous block of elements in the array.

**Example 1:**

**Input:** nums = [3,1,4,2], p = 6

**Output:** 1

**Explanation:** The sum of the elements in nums is 10, which is not divisible by 6. We can remove the subarray [4], and the sum of the remaining elements is 6, which is divisible by 6.

**Example 2:**

**Input:** nums = [6,3,5,2], p = 9

**Output:** 2

**Explanation:** We cannot remove a single element to get a sum divisible by 9. The best way is to remove the subarray [5,2], leaving us with [6,3] with sum 9.

**Example 3:**

**Input:** nums = [1,2,3], p = 3

**Output:** 0

**Explanation:** Here the sum is 6. which is already divisible by 3. Thus we do not need to remove anything.

**Example 4:**

**Input:** nums = [1,2,3], p = 7

**Output:** -1

**Explanation:** There is no way to remove a subarray in order to get a sum divisible by 7.

**Example 5:**

**Input:** nums = [1000000000,1000000000,1000000000], p = 3

**Output:** 0

**Constraints:**

* 1 <= nums.length <= 105
* 1 <= nums[i] <= 109
* 1 <= p <= 109