# Problem 2113: Elements in Array After Removing and Replacing Elements

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 70.21%
**Paid Only:** Yes
**Tags:** Array

## Problem Description

You are given a **0-indexed** integer array `nums`. Initially on minute `0`, the array is unchanged. Every minute, the **leftmost** element in `nums` is removed until no elements remain. Then, every minute, one element is appended to the **end** of `nums`, in the order they were removed in, until the original array is restored. This process repeats indefinitely.

* For example, the array `[0,1,2]` would change as follows: `[0,1,2] -> [1,2] -> [2] -> [] -> [0] -> [0,1] -> [0,1,2] -> [1,2] -> [2] -> [] -> [0] -> [0,1] -> [0,1,2] -> ...`

You are also given a 2D integer array `queries` of size `n` where `queries[j] = [timej, indexj]`. The answer to the `jth` query is:

* `nums[indexj]` if `indexj < nums.length` at minute `timej` * `-1` if `indexj >= nums.length` at minute `timej`

Return _an integer array`ans` of size _`n` _where_`ans[j]`_is the answer to the_`jth` _query_.

**Example 1:**

**Input:** nums = [0,1,2], queries = [[0,2],[2,0],[3,2],[5,0]] **Output:** [2,2,-1,0] **Explanation:** Minute 0: [0,1,2] - All elements are in the nums. Minute 1: [1,2] - The leftmost element, 0, is removed. Minute 2: [2] - The leftmost element, 1, is removed. Minute 3: [] - The leftmost element, 2, is removed. Minute 4: [0] - 0 is added to the end of nums. Minute 5: [0,1] - 1 is added to the end of nums. At minute 0, nums[2] is 2. At minute 2, nums[0] is 2. At minute 3, nums[2] does not exist. At minute 5, nums[0] is 0.

**Example 2:**

**Input:** nums = [2], queries = [[0,0],[1,0],[2,0],[3,0]] **Output:** [2,-1,2,-1] Minute 0: [2] - All elements are in the nums. Minute 1: [] - The leftmost element, 2, is removed. Minute 2: [2] - 2 is added to the end of nums. Minute 3: [] - The leftmost element, 2, is removed. At minute 0, nums[0] is 2. At minute 1, nums[0] does not exist. At minute 2, nums[0] is 2. At minute 3, nums[0] does not exist.

**Constraints:**

* `1 <= nums.length <= 100` * `0 <= nums[i] <= 100` * `n == queries.length` * `1 <= n <= 105` * `queries[j].length == 2` * `0 <= timej <= 105` * `0 <= indexj < nums.length`

## Code Snippets

**C++:**

```
class Solution {
public:
vector<int> elementInNums(vector<int>& nums, vector<vector<int>>& queries) {

}
};
```

**Java:**

```
class Solution {
public int[] elementInNums(int[] nums, int[][] queries) {

}
}
```

**Python3:**

```
class Solution:
def elementInNums(self, nums: List[int], queries: List[List[int]]) ->
List[int]:
```