# 题目

给你一个整数数组 nums ，其中 nums[i] 要么是一个正整数，要么是 -1 。我们需要为每个 -1 找到相应的正整数，我们称之为最后访问的整数。

为了达到这个目标，定义两个空数组：seen 和 ans。

从数组 nums 的头部开始遍历。

1、如果遇到正整数，把它添加到seen的头部。

2、如果遇到 -1，则设k是到目前为止看到的连续-1的数目(包括当前-1)，

如果 k 小于等于 seen 的长度，把 seen 的第 k 个元素添加到 ans。

如果 k 严格大于 seen 的长度，把 -1 添加到 ans。

请你返回数组 ans。

示例 1：

输入：nums = [1,2,-1,-1,-1]

输出：[2,1,-1]

解释： 开始时 seen = [] 且 ans = []。

1、处理 nums[0]：nums 中的第一个元素是 1。我们将其放在 seen 的前面。现在，seen == [1]。

2、处理 nums[1]：下一个元素是 2。我们将其放在 seen 的前面。现在，seen == [2, 1]。

3、处理 nums[2]：下一个元素是 -1。这是 -1 的第一次出现，所以 k == 1。我们找到 seen 中的第一个元素，把 2 添加到 ans。现在，ans == [2]。

4、处理 nums[3]：又一个 -1。这是 -1 的第二次出现，所以 k == 2。seen 中的第二个元素是 1，所以我们把 1 添加到 ans。现在，ans == [2, 1]。

5、处理 nums[4]：又一个 -1。第三次出现，让 k = 3。然而，seen 中只有两个元素（[2, 1]）。因为 k 比 seen 中的元素数量更大，我们把 -1 添加到 ans。最终，ans == [2, 1, -1]。

示例 2：

输入：nums = [1,-1,2,-1,-1]

输出：[1,2,1]

解释： 开始时 seen = [] 且 ans = []。

1、处理 nums[0]：nums 中的第一个元素是 1。我们将其放在 seen 的前面。现在，seen == [1]。

2、处理 nums[1]：下一个元素是 -1。这是 -1 的第一次出现，所以 k == 1。我们找到 seen 中的第一个元素，即 1。把 1 添加到 ans。现在，ans == [1]。

3、处理 nums[2]：下一个元素是 2。我们将其放在 seen 的前面。现在，seen == [2, 1]。

4、处理 nums[3]：下一个元素是 -1。这个 -1 与 第一个 -1 不连续，因为中间有个 2。因此，k 重置为 1。seen 中的第一个元素是 2，所以我们把 2 添加到 ans。现在，ans == [2, 2]。

5、处理 nums[4]：又一个 -1。它与前一个 -1 相邻，所以 k == 2。seen 中的第 2 个元素是 1。把 1 添加到 ans。最终，ans == [1, 2, 1]。

提示：

1 <= nums.length <= 100

nums[i] == -1 或 1 <= nums[i] <= 100

# 分析