126.You are given two integer arrays nums1 and nums2 of sizes n and m, respectively. Calculate the following values: answer1 : the number of indices i such that nums1[i] exists in nums2. answer2 : the number of indices i such that nums2[i] exists in nums1 Return [answer1,answer2].

Example 1:
Input: nums1 = [2,3,2], nums2 = [1,2]
Output: [2,1]
Explanation:

Example 2:
Input: nums1 = [4,3,2,3,1], nums2 = [2,2,5,2,3,6]
Output: [3,4]
Explanation:
The elements at indices 1, 2, and 3 in nums1 exist in nums2 as well. So answer1 is 3.
The elements at indices 0, 1, 3, and 4 in nums2 exist in nums1. So answer2 is 4.

PROGRAM:-

```
def count_common_indices(nums1, nums2):

   set_nums2 = set(nums2)

   set_nums1 = set(nums1)



   answer1 = sum(1 for num in nums1 if num in set_nums2)

   answer2 = sum(1 for num in nums2 if num in set_nums1)



   return [answer1, answer2]



# Example 1

nums1_1 = [2, 3, 2]

nums2_1 = [1, 2]

output1 = count_common_indices(nums1_1, nums2_1)

print(f"Input: nums1 = {nums1_1}, nums2 = {nums2_1}\nOutput: {output1}")



# Example 2

nums1_2 = [4, 3, 2, 3, 1]

nums2_2 = [2, 2, 5, 2, 3, 6]
```

output2 = count_common_indices(nums1_2, nums2_2)

print(f"Input: nums1 = {nums1_2}, nums2 = {nums2_2}\nOutput: {output2}")

OUTPUT:-

```
Input: nums1 = [2, 3, 2], nums2 = [1, 2]
Output: [2, 1]
Input: nums1 = [4, 3, 2, 3, 1], nums2 = [2, 2, 5, 2, 3, 6]
Output: [3, 4]

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n)