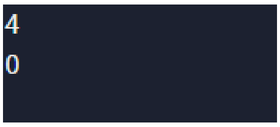
129 . Given a 0-indexed integer array nums of length n and an integer k, return *the number of pairs* (i, j) *where* 0 <= i < j < n, *such that* nums[i] == nums[j] *and* (i * j) *is divisible by* k.
Example 1:
Input: nums = [3,1,2,2,2,1,3], k = 2
Output: 4
Example 2:
Input: nums = [1,2,3,4], k = 1
Output: 0

AIM: To find the interger array nums of length n and an integer

PROGRAM:

```
def count_pairs(nums, k):
    from collections import defaultdict

    n = len(nums)
    count = defaultdict(int)
    position = defaultdict(list)

    for idx, num in enumerate(nums):
        count[num] += 1
        position[num].append(idx)

    result = 0

    for num in count:
        freq = count[num]
        indices = position[num]

        for i in range(len(indices)):
            for j in range(i + 1, len(indices)):
                if (indices[i] * indices[j]) % k == 0:
                    result += 1

    return result

nums1 = [3, 1, 2, 2, 2, 1, 3]
k1 = 2
print(count_pairs(nums1, k1))

nums2 = [1, 2, 3, 4]
k2 = 1
print(count_pairs(nums2, k2))
```

OUTPUT: 

```
4
0
```

TIME COMPLEXITY: O(n^2)