1.4 COUNT NUMBER OF PAIRS

Question:

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.

AIM:

To find the number of pairs (i, j) in an array such that:

1. 0 <= i < j < n
2. nums[i] == nums[j]
3. (i * j) is divisible by k.

ALGORITHM:

1. Initialize count = 0.
2. Iterate over all possible pairs (i, j) where i < j:
   - Check if nums[i] == nums[j].
   - Check if (i * j) % k == 0.
   - If both conditions hold, increment count.
3. Return count at the end.

PROGRAM:

```
def count_valid_pairs(nums, k):
    count = 0
    for i in range(len(nums)):
        for j in range(i+1, len(nums)):
            if nums[i] == nums[j] and (i * j) % k == 0:
                count += 1
    return count

nums = list(map(int, input("Enter nums: ").split()))
k = int(input("Enter k: "))
print("Number of valid pairs:", count_valid_pairs(nums, k))
```

Input:

nums = [3,1,2,2,2,1,3], k = 2

Output:

```
Enter nums: 3 1 2 2 2 1 3
Enter k: 2
Number of valid pairs: 4
>>>
```

RESULT:

Thus the program is successfully executed, and the output is verified.

PERFORMANCE ANALYSIS:

Time Complexity:

- Checking all pairs = O(n^2)
- Each check is O(1)
- Total: O(n^2)

Space Complexity:

- Only a counter variable → O(1)