**1.4** 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:

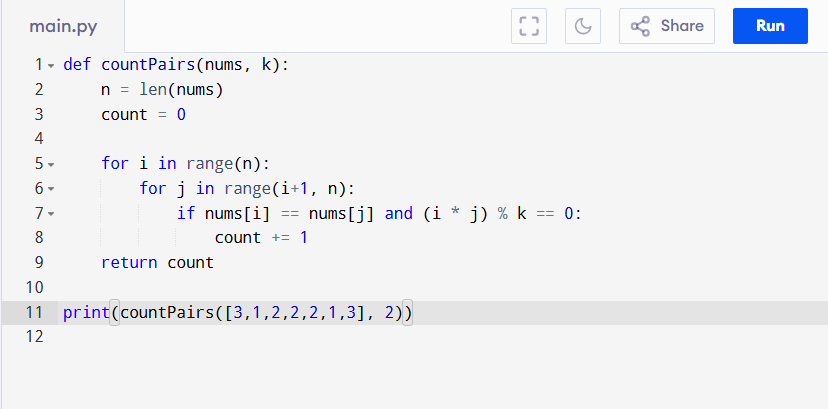
• Check if nums[i] == nums[j].

• Check if (i \* j) % k == 0.

• If both conditions hold, increment count.

3. Return count at the end.

**PROGRAM:**



Input:

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

Output:

A screenshot of a computer

AI-generated content may be incorrect.

**RESULT:**

Thus the program to count number of pairs 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)