Given a **0-indexed** integer array nums and an integer d, return *the number of triplets* (i, j, k) *such that* i < j < k *and* (nums[i] + nums[j] + nums[k]) % d == 0.

**Example 1:**

Input: nums = [3,3,4,7,8], d = 5  
Output: 3  
Explanation: The triplets which are divisible by 5 are: (0, 1, 2), (0, 2, 4), (1, 2, 4).  
It can be shown that no other triplet is divisible by 5. Hence, the answer is 3.

**Example 2:**

Input: nums = [3,3,3,3], d = 3  
Output: 4  
Explanation: Any triplet chosen here has a sum of 9, which is divisible by 3. Hence, the answer is the total number of triplets which is 4.

**Example 3:**

Input: nums = [3,3,3,3], d = 6  
Output: 0  
Explanation: Any triplet chosen here has a sum of 9, which is not divisible by 6. Hence, the answer is 0.

**Constraints:**

* 1 <= nums.length <= 1000
* 1 <= nums[i] <= 109
* 1 <= d <= 109