Given an integer array nums and an integer k, return *the number of non-empty subarrays that have a sum divisible by* k.

A subarray is a contiguous part of an array.

Example 1:

Input: nums = [4,5,0,-2,-3,1], k = 5
Output: 7
Explanation: There are 7 subarrays with a sum divisible by k = 5:
[4, 5, 0, -2, -3, 1], [5], [5, 0], [5, 0, -2, -3], [0], [0, -2, -3], [-2, -3]

Example 2:

Input: nums = [5], k = 9
Output: 0

Constraints:

- **1 <= nums.length <= $3 * 10^4$**
- **-10^4 <= nums[i] <= 10^4**
- **2 <= k <= 10^4**

Solution:

```
class Solution {
    public int subarraysDivByK(int[] nums, int k) {
        int count=0,sum=0;
        int []ans=new int[k];
        ans[0]=1;
        for(int i=0;i<nums.length;i++){
            sum=(sum+nums[i]%k+k)%k;
            count+=ans[sum];
            ans[sum]++;
        }
        return count;
    }
}
```