Qn Link : <https://leetcode.com/problems/subarray-sums-divisible-by-k/description/>

Step 1 : The same logic that is used for modulo .

Step 2 : when a subarray is divisible by k , then it modulo value should be zero .

Step 3 : We already know that , if a running sum value is repeated after the modulo , then the elements between them is modulo of K.

Step 4 : We here , need to handle an extra case , such that when sum < 0 , add K to it and then find the value in the array.

Step 5 : Check whether sum % k is exists in the array , then add the count to the answer.

Step 6 : Put the value of sum % k in the array .

class Solution {

    public int subarraysDivByK(int[] nums, int k) {

        HashMap<Integer, Integer> hm = new HashMap<>();

        int sum = 0;

        int count = 0;

        hm.put(0, 1);

        for(int i : nums) {

            sum = (i+sum)%k;

            if(sum < 0) sum += k;

            if(hm.containsKey(sum)) count += hm.get(sum);

            hm.put(sum, hm.getOrDefault(sum, 0)+1);

        }

        return count;

    }

}