**Given an array of integers nums and an integer k, return *the total number of subarrays whose sum equals to* k.**

**A subarray is a contiguous non-empty sequence of elements within an array.**

**Example 1:**

**Input: nums = [1,1,1], k = 2**
**Output: 2**

**Example 2:**

**Input: nums = [1,2,3], k = 3**
**Output: 2**
**Constraints:**

- **1 <= nums.length <= 2 * $10_4$**
- **-1000 <= nums[i] <= 1000**
- **-$10_7$ <= k <= $10_7$**

## Solution:

```
class Solution {
    public int subarraySum(int[] nums, int k) {
        int n=nums.length,ans=0;
        int[] sum = new int[n];
        sum[0]=nums[0];
        for(int i=1;i<n;i++)    sum[i]+=sum[i-1]+nums[i];
        HashMap<Integer,Integer> map = new HashMap<>();
        map.put(0,1);
        for(int i=0;i<n;i++){
            if(map.containsKey(sum[i]-k)){
                ans += map.get(sum[i]-k);
            }
            map.put(sum[i],map.getOrDefault(sum[i],0)+1);
        }
        return ans;
    }
}
```