Qn Link : <https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/description/>

Observation :

* | Nums[i] – nums[j] |= k
* Nums[i] – nums[j] = k and nums[j] – nums[i] = k

The eqn can be re written as ,

* Nums[i] = k – nums[j]
* Nums[i] = k + nums[j]
* Edge case : if req1 & req2 is zero , then count only one time

class Solution {

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

        int cnt = 0;

        int n = nums.length;

        Map<Integer , Integer> map = new HashMap<>();

        for(int i = 0 ; i < n ; i++){

            int req1 = nums[i] + k;

            int req2 = nums[i] - k;

            if(map.containsKey(req1)){

                cnt+= map.get(req1);

            }

            if(k != 0 && map.containsKey(req2)){

                cnt+= map.get(req2);

            }

            map.put(nums[i] , map.getOrDefault(nums[i] , 0) + 1);

        }

        return cnt;

    }

}