**Subarrays with sum K**

Given an unsorted array of integers, find the number of continuous subarrays having sum exactly equal to a given number k.

**Example 1:**

**Input:**

N = 5

Arr = {10 , 2, -2, -20, 10}

k = -10

**Output:** 3

**Example 2:**

**Input:**

N = 6

Arr = {9, 4, 20, 3, 10, 5}

k = 33

**Output:** 2

**Step 1 :** Initialize an empty map with {0 , 1} **.**

**Step 2 :** for each iteration , check whether sum – k exists in the map , it is so , then add the cnt to the answer.

**Step 3 :** put then answer and add the occurrence in the map.

class Solution

{

static int findSubArraySum(int nums[], int n, int k)

{

// code here

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

map.put(0 , 1);

int cnt = 0;

int sum = 0;

for(int i= 0 ; i< n; i++){

sum += nums[i];

int req = sum - k;

if(map.containsKey(req)){

cnt+= map.get(req);

}

// System.out.println(map+" "+cnt);

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

}

return cnt;

}

}