**[Continuous Subarray Sum](https://leetcode.com/problems/continuous-subarray-sum/)**

**package** leetcodeMedium;

**import** java.util.HashMap;

**public** **class** ContinuousSubarraySum {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[] nums = {23, 2, 4, 6, 7};

System.***out***.println(*checkSubarraySum*(nums , 6));

}

**public** **static** **boolean** checkSubarraySum(**int**[] nums, **int** k) {

**if**(nums == **null** || nums.length == 0) {

**return** **false**;

}

**int** sum = 0;

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

map.put(0 , -1);

**for**(**int** i = 0 ; i < nums.length ; i++) {

sum += nums[i];

**if**(k != 0) {

sum = sum % k;

}

**if**(!map.containsKey(sum)) {

map.put(sum, i);

}

**else** {

**if**(i - map.get(sum) > 1) {

**return** **true**;

}

}

}

**return** **false**;

}

}

Time complexity : O(n) , Number of elements in array

Space Complexity : O(n) , Number of elements in array