Question: https://leetcode.com/problems/continuous-subarray-sum/

So the idea is to maintain a running sum and store running sum%k as key and the position as value.

Now if we see that the key previously existed earlier at a gap greater than that means the subarray whose sum is multiple of k exists.

Code:  
class Solution {

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

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

map.put(0, -1);

int runningSum = 0;

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

runningSum += nums[i];

if (k != 0) runningSum %= k;

Integer prev = map.get(runningSum);

if (prev != null) {

if (i - prev > 1) return true;

}

else map.put(runningSum, i);

}

return false;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>