**Chủ nhật, ngày 09 tháng 06 năm 2024**

Cho mảng A có n phần tử (n<=1e5) và một số nguyên k (k<=1e4). Hãy xác định xem mảng A có bao nhiêu dãy con liên tiếp có tổng chia hết cho k

**Input**

Dòng đầu là số bộ test T (T<=10)

Mỗi bộ test gồm 2 dòng

* Dòng đầu tiên là số n là số phần tử mảng a
* Dòng thứ hai là n số của mảng a (abs(a[i])<=1e4)

**Output**

Với mỗi bộ test, hãy in ra số dãy con liên tiếp thoả mãn tương ứng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  6 5  4 5 0 -2 -3 1  3 2  -1 2 9  4 6  2 -2 2 -4 | 7  2  2 |

**Giải thích test:**

Test 1: Có 7 dãy con thoả mãn là:

[4, 5, 0, -2, -3, 1], [5], [5, 0], [5, 0, -2, -3], [0], [0, -2, -3], [-2, -3]

**Lời giải**

#include<bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin>>t;

    while(t--){

        int n, k;

        cin>>n>>k;

        int a[n];

        unordered\_map<int, int>mp;

        for(int i = 0;i<n;i++) cin>>a[i];

        int tong = 0, cnt = 0; mp[0] = 1;

        for(int i = 0;i<n;i++){

            tong+=a[i];

            int mod = tong%k;

            if (mod <0) mod+=k;

            cnt+=mp[mod];

            mp[mod]++;

        }

        cout<<cnt<<endl;

    }

}

**Phân tích**

Gọi mp[s] = 5, tức là có 5 index i thoả mãn prefix[i]%k có kết quả là s

Tổng đến index I hiện tại có số dư là tổng%k, vậy ta phải tìm xem, trước I có bao nhiêu index j < I sao cho prefix[j]%k = tong%k 🡪Nhiệm vụ của mp[tong%k]

Hiển nhiên, lấy prefix[i] – prefix[j] sẽ ra tổng từ index I + 1 đến index j, và 2 số dư sẽ triệt tiêu nhau

***Xử lý ngoại lệ***

1. **Số dư âm khi lấy a - b**

Ví dụ a = 28, b = -12, k = 5

a = 5 \* k + 3, b = (-2) \* k + (-2).

Nếu để nguyên số dư đưa vào mp, rõ ràng đây là 2 key khác nhau, không tăng thêm được biến cnt

Tuy nhiên, a – b = 40 lại chia hết cho 5

Cho nên ta phải chuyển số dư của -12 khia chia cho 5 về số dư dương

Hiểu nôm na là: Giống như kiểu 12 = 5 \* 3 – 3

Trừ 3 ở đây có nghĩa là: Thiếu 3 cái nữa thì mới đủ để tạo thành bộ thứ 3.

Như vậy, bộ lẻ này đang có 2.

Vì vậy, khi số dư mà âm thì số dư dương = số dư âm + k

1. **Lí do mp[0] = 1**

Tức là ban đầu khi chưa xét thì là dãy rỗng, không tồn tại index nào cả.

Nếu không có mp[0] = 1 thì ví dụ đến index I mà tổng chia hết cho k thì sẽ không update đc