【华为OD机考 统一考试机试C卷】寻找连续区间/数组连续和（C++ Java JavaScript Python C语言）

华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，C卷真题已基本整理完毕
抽到原题的概率为2/3到3/3，也就是最少抽到两道原题。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。
另外订阅专栏还可以联系笔者开通在线 OJ 进行刷题，提高刷题效率。
真题目录：华为OD机考机试 真题目录（C卷 + D卷 + B卷 + A卷） + 考点说明
专栏：2023华为OD机试(B卷+C卷+D卷)（C++JavaJSPy）
华为OD面试真题精选：华为OD面试真题精选
在线OJ：点击立即刷题，模拟真实机考环境

题目描述

给定一个含有N个正整数的数组, 求出有多少个连续区间（包括单个正整数），它们的和大于等于x。

输入描述

第一行两个整数N x（0 < N <= 100000, 0 <= x <= 10000000)

第二行有N个正整数（每个正整数小于等于100)。

输出描述

输出一个整数，表示所求的个数。

用例1

输入

```
3 7

```

```
3 4 7
```

输出

```
4
```

> 第一行的3表示第二行数组输入3个数，第一行的7是比较数，用于判断连续数组是否大于该数；组合为 3 + 4; 3 + 4 + 7; 4 + 7; 7; 都大于等于指定的7；所以共四组.

用例2

输入

```
10 10000
1 2 3 4 5 6 7 8 9 10
```

输出

```
0
```

> 所有元素的和小于10000，所以返回0。

C++

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, x;
    cin >> n >> x;

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

    int left = 0; // 滑动窗口的左端点

```

```
    int right = 0; // 滑动窗口的右端点
    int count = 0; // 记录连续区间个数
    int sum = 0; // 记录当前区间的和

    while (right < n) {
        sum += nums[right];

        while (sum >= x) {
            // 如果当前区间和大于等于x，那么以left为起点的所有连续区间都符合要求
            count += n - right;
            sum -= nums[left];
            left++;
        }

        right++;
    }

    cout << count << endl;

    return 0;
}
```

javaScript

```
const readline = require('readline');

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

let n = 0;
let x = 0;
let nums = [];

rl.on('line', (line) => {
  if (!n) {
    [n, x] = line.trim().split(' ').map(Number);
  } else {
```

```
    nums = line.trim().split(' ').map(Number);

    let left = 0; // 滑动窗口的左端点
    let right = 0; // 滑动窗口的右端点
    let count = 0; // 记录连续区间个数
    let sum = 0; // 记录当前区间的和

    while (right < n) {
      sum += nums[right];

      while (sum >= x) {
        // 如果当前区间和大于等于x，那么以left为起点的所有连续区间都符合要求
        count += n - right;
        sum -= nums[left];
        left++;
      }

      right++;
    }

    console.log(count);
  }
});
```

python

```
import sys

n, x = map(int, sys.stdin.readline().split())

nums = list(map(int, sys.stdin.readline().split()))

left = 0
right = 0
count = 0
sum = 0

while right < n:
    sum += nums[right]

```

```
    while sum >= x:
        count += n - right
        sum -= nums[left]
        left += 1

    right += 1

print(count)
```

Java

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int x = scanner.nextInt();

        int[] nums = new int[n];
        for (int i = 0; i < n; i++) nums[i] = scanner.nextInt();

        int left = 0; // 滑动窗口的左端点
        int right = 0; // 滑动窗口的右端点
        int count = 0; // 记录连续区间个数
        int sum = 0; // 记录当前区间的和

        while (right < n) {
            sum += nums[right];

            while (sum >= x) {
                // 如果当前区间和大于等于x，那么以left为起点的所有连续区间都符合要求
                count += n - right;
                sum -= nums[left];
                left++;
            }

            right++;
        }

```

```
        System.out.println(count);
    }
}
```

C语言

```c
#include <stdio.h>

int main() {
    int n, x;
    scanf("%d %d", &n, &x); // 读取N和x

    int nums[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]); // 读取数组中的正整数
    }

    int left = 0; // 滑动窗口的左端点
    int right = 0; // 滑动窗口的右端点
    int count = 0; // 记录连续区间个数
    int sum = 0; // 记录当前区间的和

    // 遍历数组，使用滑动窗口计算连续区间的和
    while (right < n) {
        sum += nums[right]; // 将右端点的值加到区间和中

        // 当区间和大于等于x时，移动左端点
        while (sum >= x) {
            count += n - right; // 以left为起点的所有连续区间都符合要求
            sum -= nums[left]; // 移动左端点，区间和减去左端点的值
            left++;
        }

        right++; // 移动右端点
    }

    printf("%d\n", count); // 输出符合条件的连续区间个数

    return 0;
}
```

完整用例

用例1

```
3 7
3 4 7
```

用例2

```
5 10
1 2 3 4 5
```

用例3

```
4 5
1 2 3 4
```

用例4

```
6 15
1 2 3 4 5 6
```

用例5

```
8 20
2 4 6 8 10 12 14 16
```

用例6

```
10 100
10 20 30 40 50 60 70 80 90 100
```

用例7

```
7 50
10 20 30 40 50 60 70
```

用例8

```
15 80
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
```

用例9

```
12 60
12 24 36 48 60 72 84 96 108 120 132 144
```

用例10

```
9 70
7 14 21 28 35 42 49 56 63
```

文章目录

用例7
用例8
用例9
用例10