

Counting

Statement

Today our Grandpa is hosting a party with a very close friend of him: Practical Exam. And guess what, the party is an algorithm party!!! Practical Exam have asked Uncle Grandpa two questions, and the first one is as follows:

“You are given an array consisting of n non-negative integers a_1, a_2, \dots, a_n and an integer k . Count the number of continuous subarrays that has the sum of its elements greater than or equal to k . ”

Do you think our Grandpa is able to solve this problem? Of course he can, he solved it in just 45 minutes. Let’s solve it faster than him!

Constraints

The following constraints apply for all subtasks.

- $1 \leq n \leq 10^5$
- $0 \leq k \leq 10^{18}$
- $0 \leq a_i \leq 10^9$

Input

The first line contains two integers n, k — the length of the array and the parameter k .

The second line contains n integers a_i — the elements of the array.

Output

Print a single integer — the answer to the problem.

Examples

Sample Input	Expected Output
<pre>5 8 4 5 3 1 2</pre>	<pre>7</pre>
<pre>6 6 4 1 1 3 3 4</pre>	<pre>11</pre>

For the 1st example, the array a is $[4, 5, 3, 1, 2]$ with $k = 8$. The 7 continuous subarray of array a that has sum ≥ 8 are:

1. $a[1 \dots 2]$ ($sum = 9$)
2. $a[1 \dots 3]$ ($sum = 12$)
3. $a[2 \dots 3]$ ($sum = 8$)
4. $a[1 \dots 4]$ ($sum = 13$)
5. $a[2 \dots 4]$ ($sum = 9$)
6. $a[1 \dots 5]$ ($sum = 15$)
7. $a[2 \dots 5]$ ($sum = 11$)

Subtasks

You may choose to solve any one of the following subtasks.

This accounts for 70% of your score for this problem.

- **Subtask 1 (60%):**

To obtain full credit for this subtask, your solution should run in $O(n^3)$ time.

- **Subtask 2 (80%):**

To obtain full credit for this subtask, your solution should run in $O(n^2)$ time.

- **Subtask 3 (100%):**

To obtain full credit for this subtask, your solution should run in $O(n \log n)$ time.

- **Subtask 4 (110%):**

To obtain full credit for this subtask, your solution should run in $O(n)$ time.

Marking Scheme

- Correctness and Efficiency (70%)
 - Refer to the Subtasks section.
- Programming Style (30%)
 - Purpose of methods and statements (5%)
 - Pre- and post- conditions (5%)
 - Modularity (5%)
 - Scoping (5%)
 - Meaningful Identifiers (4%)
 - Variable Naming (1%)
 - Indentation (5%)

Notes

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided.

Skeleton File

You are given the skeleton file `Counting.java`. You should see the following contents when you open the file:

```
/**
 * Name      :
 * Matric. No :
 */

import java.util.*;

public class Counting {
    private void run() {
        // implement your "main" method here
    }

    public static void main(String args[]) {
        Counting runner = new Counting();
        runner.run();
    }
}
```

```
}  
}
```