EXPRIMIDOR

Kolya is going to make orange juice. He has n oranges sizes a1, a2, ..., an. Kolya lay them in the blender in a fixed order, from orange size A1 then A2 size orange and so on. To put in the blender the orange should have a size not greater than B, so if Kolya sees an orange that is strictly greater than B strip away and continues to the next.

The juicer has a special section to collect waste. Kolya overflows if the sum squeezing oranges size is greater than d. When Kolya happens empty the waste section (not even if more than oranges) and continues to squeeze the juice. How many times have to empty the waste section?

SPANISH VERSION

Input

The first line of the input contains three integers n, B and D (1 ≤ n ≤ 100000, 1 ≤ B ≤ d ≤ 1000000) - the number of oranges, the maximum size of the orange that fits in the blender and the d value, which determines the condition when the section waste must be emptied.
The second line contains n integers a1, a2, ..., n (1 ≤ AI ≤ 1000000) - oranges sizes listed in the order and Kolya will try to put in blender.

Output

Imprimir un entero - el número de veces que Kolya tendrá que vaciar la sección de desperdicio

Example

Input:

2 7 10

5 6

Output:

1

Input:

1 5 10

7

Output:

0

Input:

3 10 10

5 7 7

Output:

1

Input:

1 1 1

1

Output:

0

Note
In the first example, Kolya squeeze the juice of two oranges and empty the waste section later.
In the second example, the orange will not fit into the juicer so that Kolya will not have any juice at all.

In the third example, Kolya squeeze the juice of two oranges and empty the waste section later, then squeezes the other orange.