B. Candy

Problem Name	Candy
Time Limit	3 seconds
Memory Limit	1 gigabyte

In the ancient city of Ica, there is said to be a palace with wealth beyond imagination. Inside, there is a corridor with N boxes of candy from all over the world. Travellers passing by can take as much candy as they want, provided that they pay its weight in gold.

The boxes of candy are numbered 0 to $N-1$ from left to right. In box i, there are a_i units of candy left, where a_i is a non-negative integer.

As the guardian of the palace, you would like to move the boxes around so that boxes with a lot of candy end up closer to the entrance.

You are given the array $a_0, a_1, \ldots, a_{N-1}$, as well as the numbers F and T. In a single operation, you are allowed to swap two **adjacent** elements of $a_0, a_1, \ldots, a_{N-1}$. What is the minimum number of operations required so that the first F elements of the array sum to at least T?

Input

The first line of the input contains three integers, N, F, and T.

The second line of the input contains N integers $a_0, a_1, \ldots, a_{N-1}$.

Output

If it is impossible to achieve the objective using the operations, print "NO".

Otherwise, print a single integer, the minimum number of operations.

Constraints and Scoring

- $1 \le N \le 100$.
- $1 \le F \le N$.
- $0 \le T \le 10^{11}$.

- $0 \leq a_i \leq 10^9$ for $i = 0, 1, \ldots, N-1$.

Note: The numbers in the input may not fit in a 32-bit integer, so be aware of overflows if you are using C++.

Your solution will be tested on a set of test groups, each worth a number of points. Each test group contains a set of test cases. To get the points for a test group, you need to solve all test cases in the test group.

Group	Score	Limits
1	6	$N \leq 2$ and $a_i \leq 100$ for $i = 0, 1, \ldots, N-1$ and $T \leq 10^9$
2	19	$a_i \leq 1$ for $i = 0, 1, \ldots, N-1$
3	16	$N \leq 20$
4	30	$a_i \leq 100$ for $i = 0, 1, \ldots, N-1$
5	29	No additional constraints

Example

In the first sample test case, the first two elements should sum to at least 27. This can be achieved by a single swap of two adjacent elements: swap the 4 and 20. After this swap, the array becomes `10 20 4 6 3 3`, and indeed the first two elements sum to $10 + 20 = 30 \geq 27$.

In the second sample test case, the 0 must move all the way to the end of the array; this takes three swaps.

In the third sample test case, it is impossible to make the first two elements sum to at least 100; the best we can do is $60 + 30 = 90$.

Input	Output
6 2 27 10 4 20 6 3 3	1
6 5 5000000000 1000000000 1000000000 0 1000000000 1000000000 1000000000	3
3 2 100 20 30 60	NO
1 1 100 100	0