Halloween Candies (`halloween`)

Marco prepared a big basket with M candies for the usual "trick-or-treat" that is going to take place in his district during Halloween. He placed this basket on his porch, in front of the door, so that kids can take candies without having to ring the doorbell.

In this district, things are very regular and the N kids who live here are numbered from 0 to $N-1$. Some kids are completely selfless and will not take any candies (to leave more candies for other kids to enjoy) while some other kids are greedy and will take one or more candies whenever they approach a house.

Figure 1: Kids approaching a house to get candies.

Marco knows that the kids will approach his house in order, from the 0-th kid to the $(N-1)$-th kid, and that they will repeatedly come back to his house (after visiting the other houses in the district) in the same order, until there are no candies left.

Help Marco calculate which kid is going to take the last candy in the basket!

☞ Among the attachments of this task you may find a template file `halloween.*` with a sample incomplete implementation.

Input

The first line contains two integers N and M. The second line contains N integers L_i, the number of candies that each kid will take during his/her turn.

Output

You need to write a single line with an integer: the index (from 0 to $N-1$) of the kid that is going to take the very last candy from the basket.

Constraints

- $1 \leq N \leq 100\,000$.
- $1 \leq M \leq 10^{18}$.
- $0 \leq L_i \leq 10^9$ for each $i = 0 \ldots N-1$.
- There is at least one greedy kid i (with $L_i > 0$).

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points)   Examples.

– **Subtask 2** (30 points) $N, M \leq 1000$.

– **Subtask 3** (20 points) $M \leq 1\,000\,000$.

– **Subtask 4** (10 points)   Every kid takes exactly 1 candy.

– **Subtask 5** (15 points)   Every kid takes at least 1 candy.

– **Subtask 6** (25 points)   No additional limitations.

Examples

input	output
5 9 1 0 2 1 1	3
2 1000000000000000000 1 1	1

Explanation

In the **first sample case**, the kid with index 3 will get the last candy.

In the **second sample case** at each turn the two kids take 1 candy each, which means that the last candy will be taken by the second kid (index 1) since M is an even number.