Vibing Network (vibingnetwork)

Your ops channel has a running joke: "If at least 67% of devices are up, the network is vibing."

You monitor a cluster of IoT devices for several minutes. For each minute, you know how many of the N devices reported as healthy. The team wants to find the **first moment** when the network becomes "vibing" and **stays that way** for K consecutive minutes.

Definition. Minute i is called **good** if at least 67% of devices are online at that minute:

$$a_i \cdot 100 \geq 67 \cdot N$$

where a_i is the number of online devices at minute i.

Task. Given N, T, K, and the values $a_1, a_2, \ldots, a_T$, find the **earliest** index i (1-based) such that minutes $i, i+1, \ldots, i+K-1$ are all good. If no such i exists, output -1.

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

Input

The first line contains three integers N, T, and K: the number of devices, the number of minutes monitored, and the required streak length, respectively.

The second line contains T integers $a_1, a_2, \ldots, a_T$: the number of online devices at each minute.

Output

Output a single integer: the earliest 1-based index i such that minutes $i, i+1, \ldots, i+K-1$ are all good, or -1 if no such index exists.

Constraints

- $1 \leq N \leq 1\,000\,000$.
- $1 \leq T \leq 200\,000$.
- $1 \leq K \leq T$.
- $0 \leq a_i \leq N$ for each $i = 1 \ldots T$.

Note: Use integer arithmetic. Do not use floating point comparisons for the 67% threshold. The condition $a_i \cdot 100 \geq 67 \cdot N$ is equivalent to $a_i \geq \left\lceil \frac{67N}{100} \right\rceil$.

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** (15 points) $T \leq 2000$.

– **Subtask 3** (20 points) $K = 1$.

– **Subtask 4** (65 points) No additional limitations.

Examples

input	output
10 6 3 6 7 8 7 7 5	2
4 5 2 3 2 3 2 3	-1

Explanation

In the **first sample case**, we have $N = 10$, so 67% of 10 is 6.7, meaning a minute is good if and only if at least 7 devices are online.

The sequence of online devices is: $[6, 7, 8, 7, 7, 5]$.

The good/bad status of each minute is: $[bad, good, good, good, good, bad]$.

We need $K = 3$ consecutive good minutes. The earliest such streak starts at minute 2 (minutes 2, 3, 4 are all good).

In the **second sample case**, we have $N = 4$. We need $a_i \cdot 100 \geq 67 \cdot 4 = 268$, so a_i must be at least 3.

The sequence is: $[3, 2, 3, 2, 3]$.

The good minutes are exactly those with $a_i = 3$ (minutes 1, 3, 5), but they never appear twice in a row. Since $K = 2$, there is no valid streak, so the answer is -1.