Task: ZOO
Zoomies

Dexter the Cat has the Zoomies. Again. He has them every day and it's all very exhausting. Before the Zoomies, he gathers **energy** by eating and napping. He needs to exactly plan *how much energy* he needs to gather before today's Zoomies.

During the Zoomies Dexter runs around and jumps all over the place. An ignorant human might describe those jumps as "random", but Dexter always has a plan. Today, he plans to do n jumps of varying difficulty in a specific order maximising the spectacle. He also estimated how much **energy** each jump will cost him, given as e_i for each jump $1 \leq i \leq n$.

After each jump, Dexter may decide to take a short **break** to catch his breath. A break restores his energy, but each time by 1 unit less than before. So if he starts with x energy then after the first break he recovers to $x-1$, after the second to $x-2$, and so on, until he reaches 0 and the Zoomies are over.

Having many breaks makes Zoomies much less fun, so Dexter plans to do at most k breaks today. Given the difficulty levels of the jumps help Dexter decide **what is the minimal starting energy** he needs to complete all n jumps in order with at most k breaks.

Input

In the first line of standard input there are two integers n, k, the length of the planned jump sequence and the upper limit on the number of breaks.

The second line contains a sequence of n positive integers, $e_1, ..., e_n$, describing the energy costs of all jumps.

Output

Your program should write exactly one line to standard output, containing the minimal starting energy required, as described above.

Example

For the input:

```
7 2
5 3 1 9 3 2 5
```

the correct output is:

```
12
```

Dexter can perform the first three jumps for 9 energy and then take a break, which recovers his energy level to 11. He can then jump once and take a break again, recovering his energy to 10. That is exactly enough to complete the final three jumps. No smaller amount of starting energy would be enough to complete the Zoomies with only two breaks.

Additional examples

The following initial tests are also available:

- `0b` – small test for Subtask 1, $n = 15$, $k = 4$;
- `0c` – Subtask 2, $n = 250$, $k = 10$, jumps are in 10 groups of equal difficulty;
- `0d` – Subtask 3, $n = 250$, $k = 20$, each jump has energy cost 1;

- `0e` – big test, $n = 10^6$, $k = 10^5$, $e_i = i$.

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

Subtask	Limits	Points
1.	$1 \leq n \leq 250$, $0 \leq k \leq 250$, $1 \leq e_i \leq 10^3$	2
2.	$1 \leq n \leq 250$, $0 \leq k \leq 250$, $1 \leq e_i \leq 10^{12}$	2
3.	$1 \leq n \leq 10^6$, $0 \leq k \leq 10^6$, $1 \leq e_i \leq 10^{12}$, $\forall_{i,j}.e_i = e_j$	1
4.	$1 \leq n \leq 10^6$, $0 \leq k \leq 10^6$, $1 \leq e_i \leq 10^{12}$	5