

B. More Cowbell

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Kevin Sun wants to move his precious collection of n cowbells from Naperthrill to Exeter, where there is actually grass instead of corn. Before moving, he must pack his cowbells into k boxes of a fixed size. In order to keep his collection safe during transportation, he won't place more than **two** cowbells into a single box. Since Kevin wishes to minimize expenses, he is curious about the smallest size box he can use to pack his entire collection.

Kevin is a meticulous cowbell collector and knows that the size of his i -th ($1 \leq i \leq n$) cowbell is an integer S_i . In fact, he keeps his cowbells sorted by size, so $S_{i-1} \leq S_i$ for any $i > 1$. Also an expert packer, Kevin can fit one or two cowbells into a box of size S if and only if the sum of their sizes does not exceed S . Given this information, help Kevin determine the smallest S for which it is possible to put all of his cowbells into k boxes of size S .

Input

The first line of the input contains two space-separated integers n and k ($1 \leq n \leq 2 \cdot k \leq 100\,000$), denoting the number of cowbells and the number of boxes, respectively.

The next line contains n space-separated integers S_1, S_2, \dots, S_n ($1 \leq S_1 \leq S_2 \leq \dots \leq S_n \leq 1\,000\,000$), the sizes of Kevin's cowbells. It is guaranteed that the sizes S_i are given in non-decreasing order.

Output

Print a single integer, the smallest S for which it is possible for Kevin to put all of his cowbells into k boxes of size S .

Examples

input
2 1 2 5
output
7
input
4 3 2 3 5 9
output
9
input
3 2 3 5 7
output
8

Note

In the first sample, Kevin must pack his two cowbells into the same box.

In the second sample, Kevin can pack together the following sets of cowbells: $\{2, 3\}$, $\{5\}$ and $\{9\}$.

In the third sample, the optimal solution is $\{3, 5\}$ and $\{7\}$.

Codeforces Round #334 (Div. 2)

Finished

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.


[Start virtual contest](#)

→ Problem tags

[binary search](#) [greedy](#)

No tag edit access

→ Contest materials

- Announcement 
- Tutorial 