

Parallel Pipeline Scheduling

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Hiraethsoul is designing a parallel processing pipeline.

There are N sequential tasks, where the i -th task requires T_i units of time. There are K local processing cores and one cloud server available.

Since the tasks must be executed in order, the N tasks must be partitioned into M contiguous segments, where $1 \leq M \leq K$. Each segment $[L_j, R_j]$ is assigned to one core. The segments are pairwise disjoint and together cover the entire range $[1, N]$.

For each core:

- Exactly one task from its assigned segment must be offloaded to the cloud server.
- The local processing time of this core equals the total processing time of its segment minus the processing time of the offloaded task.

The cloud server can process all submitted tasks in parallel. Its total processing time is defined as the maximum processing time among all tasks offloaded to the cloud.

Formally, let:

- $Core_j$ denote the local processing time of the j -th core;
- $cloud$ denote the cloud processing time.

The total system processing time is defined as

$$\max \left(\max_{1 \leq j \leq M} Core_j, cloud \right).$$

Hiraethsoul wants to determine a partition of the tasks and the offloaded task in each segment such that the total system processing time is minimized.

Input

The first line contains two integers N and K ($1 \leq K \leq N \leq 10^6$).

The second line contains N integers T_1, T_2, \dots, T_N ($1 \leq T_i \leq 10^9$).

Output

Print a single integer — the minimum possible total completion time.

Examples

standard input	standard output
10 8 46 41 49 46 23 12 46 49 29 12	49
10 3 491 372 450 40 446 73 232 439 420 483	822