

Jobs through Pipelines

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

You need to schedule n jobs on a pipeline. The jobs form a sequence, and job i has workload a_i . The pipeline consists of m **stages** in series.

Before processing begins, you may choose an integer t ($0 \leq t \leq n$). This moves the first t jobs to the end without changing their relative order, obtaining $a_{t+1}, a_{t+2}, \dots, a_n, a_1, a_2, \dots, a_t$. Choosing $t = 0$ or $t = n$ leaves the sequence unchanged.

Next, you need to partition the resulting sequence into at most k nonempty contiguous segments, each forming a **batch**. The workload of a batch is the sum of the workloads of its jobs.

The batches then enter the pipeline in order. Each batch passes through stages $1, 2, \dots, m$ in this order, taking time equal to its workload at each stage. Each stage processes at most one batch at a time, following the batches' left-to-right order in the sequence. After finishing at one stage, a batch may wait for the next stage to become available. While it waits, it no longer occupies the previous stage, which may process the next batch. Different stages may process different batches simultaneously.

The pipeline starts processing at time 0. Choose the prefix to move and the partition into batches so that all jobs finish as early as possible.

Input

The first line contains three integers n, m, k ($1 \leq n, m \leq 5 \times 10^5$, $1 \leq k \leq n$): the number of jobs, the number of stages, and the maximum number of batches.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$), the workloads of the jobs.

Output

Print one integer: the minimum possible time at which all jobs have finished.

Example

standard input	standard output
4 3 2 3 1 4 2	20

Note

An optimal choice is to move the prefix $[3, 1, 4]$ to the end, obtaining the sequence $[2, 3, 1, 4]$, and partition it into two batches $[2, 3]$ and $[1, 4]$. Both batches have workload 5. They can start at the first stage at times 0 and 5, and finish at the third stage at times 15 and 20, respectively. Thus, all jobs finish at time 20, which is the minimum possible time.