# Tulip Bouquets (tulips)

Anna really likes tulips. She has $N$ tulips in her garden, numbered from 0 to $N-1$. The *beautiness* of tulip $i$ is $T_i$. She wants to create $K$ (non-empty) bouquets from the tulips. To do that, she starts to walk from the first tulip towards the last. At each flower, she can either

- insert it into the current bouquet, or
- finish the current bouquet and start a new one. The current tulip is added to the new bouquet.

Note that, after finishing a bouquet she won't be able to insert more tulips into it!

Figure 1: Tulips are indeed beautiful.

The *beautiness* of a bouquet is the minimum of the *beautinesses* of the tulips in it. She wants to maximize the sum of the *beautinesses* of the $K$ bouquets by partitioning the tulips optimally. Your task is to calculate this maximum value!

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

## Input

The input file consists of:

- a line containing integers $N$, $K$.
- a line containing the $N$ integers $T_0, \ldots, T_{N-1}$.

## Output

The output file must contain a single line with an integer $M$, the maximum total *beautiness* of the bouquets.

## Constraints

- $1 \leq K \leq N \leq 100\,000$.
- $1 \leq N \cdot K \leq 50\,000\,000$.
- $0 \leq T_i \leq 1\,000\,000\,000$ for each $i = 0 \ldots N-1$.

## 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** (12 points) $K \leq 2$.

– **Subtask 3** (25 points) $N \leq 16$.

– **Subtask 4** (25 points) $N \leq 500$.

– **Subtask 5** (38 points) No additional limitations.

## Examples

| input | output |
|---|---|
| 5 2<br>3 4 1 5 2 | 4 |
| 6 4<br>4 2 6 1 3 5 | 14 |

## Explanation

In the **first sample case** 3 4 | 1 5 2 is an optimal way to distribute the flowers into 2 bouquets. The total *beautiness* is $3 + 1 = 4$.

In the **second sample case** 4 2 | 6 | 1 3 | 5 is an optimal way to distribute the flowers into 4 bouquets. The total *beautiness* is $2 + 6 + 1 + 5 = 14$.