



## Load unbalancer (unbalancer)

As first task, you are required to write the worst possible software, in particular we need you to write a load (un)balancer.

In our local server we have  $N$  available workers, for each of them we know the number of tasks it has to execute. The load (un)balancer can moves any amount of available tasks from one worker to another worker. Your job is to write the load (un)balancer in a way that maximize the unbalance of the server with  $K$  possible moves available.

The unbalance of server is defined as the highest possible difference of tasks between two workers.

Write a program that, given the number of tasks and moves available, returns the maximum possible unbalance between two workers.

### Input data

**Among the attachments you can find templates for each available language that already correctly implements the management of input and output, use them!**

Your program must read the input data from the standard input.

The first line of the input contains two space-separated integers  $N$  and  $K$  representing the number of worker and the number of movements.

The next line contains  $N$  space-separated integers representing the number of tasks in each workers.

### Output data

**Among the attachments you can find templates for each available language that already correctly implements the management of input and output, use them!**

Your program must write the output data into the standard output.

The output must contains only one integer, representing the maximum difference possible between the workers after  $K$  moves.

### Scoring

For each of the test cases the program will be tested, the following constraints are met:

- $K = 1$  and  $N \leq 10$  for at least 25% of all the test cases.
- $K \leq 10$  and  $N \leq 100$  for at least 50% of all the test cases.
- $K \leq 100$  and  $N \leq 1000$  for at least 75% of all the test cases.
- $K \leq 1000$  and  $N \leq 10000$  for all the test cases.

## Examples

input	output
5 1 5 0 4 4 4	9

## Explanation

With only one possible move, we can take the tasks from worker 2 and move to the worker 0, in this way we have a maximum difference of 9 tasks between all the workers.