CyberChallenge.IT 2023 - Programming Test

Problem 3 - “The national final” [100 points]

Problem Statement

The day of the national final of CyberChallenge.IT 2023 has arrived. The best six hackers of every venue are ready to battle for the eternal glory.

The organizers' team prepared N vulnerable services for the event, that will run on each team's vulnerable machine. The *checksystem*, hosted by the organizers, has the job of checking that everything works correctly on them in every round of the competition. All the rounds have a fixed duration of T seconds.

The checksystem is a computer program with W *workers*. Every worker can execute at most one task at a time, which means that no more than W tasks can be executed at the same time. Moreover, a task executing on the checksystem can never be interrupted: if it lasts t seconds, a worker will remain busy for exactly that time. When a task finishes, the next task starts immediately, without delays between them.

For every vulnerable service, the organizers wrote a *check* routine that takes a fixed amount of time (possibly different for each service) to be executed as a task on the checksystem.

Given the number of vulnerable service N, the duration (in seconds) of a round T and N positive integers $t_1, ..., t_N$ representing the time needed (in seconds) for each *check* routine to be executed, what is the minimum number of workers W that is necessary to ensure that all the checks can be completed in the round length?

Note: the checksystem executes the tasks in the listed order. In particular, there is a common queue between workers, and when a worker finishes a task it can only start the next task that has not started yet. Moreover, tasks' times can not be splitted between workers.

Problem Details

Input

The input consists of 2 lines:

- Line 1: the numbers N and T, separated by a space.
- Line 2: N space-separated positive integers representing the length of the checks $t_1, ..., t_N$.

Output

The output must contain a single positive integer W, the minimum number of workers that is necessary to execute all the tasks.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all its testcases.

- **Subtask 1** **[30 points]**: $1 \leq N \leq 100$, $1 \leq T \leq 10^6$, $1 \leq t_1, \ldots, t_N \leq 100000$, $t_1 = t_2 = \cdots = t_N$.
- **Subtask 2** **[40 points]**: $1 \leq N \leq 1000$, $1 \leq T \leq 10^6$, $1 \leq t_1, \ldots, t_N \leq 100000$.
- **Subtask 3** **[30 points]**: $1 \leq N \leq 10^5$, $1 \leq T \leq 10^6$, $1 \leq t_1, \ldots, t_N \leq 100000$.

Examples

INPUT	OUTPUT
6 100 12 49 87 21 11 31	3

Explanation

Three workers are enough. In fact, at the beginning the first 3 tasks are assigned to the 3 workers. Worker 1 is the first to finish at time 12, so it will start the fourth task. Again, the first worker finishes a task at time 33, so it can start the 5th one. Finally, at time 44 the first worker finishes again, starting the 6th and last task. At the end, worker 1 have been busy for 75 seconds, worker 2 for 49 and worker 3 for 87.
Conversely, two workers are not enough, since the total sum of the timings is greater than 200, so it is impossible to divide them between two workers that will work for at most 100 seconds.