Problem M
Mid Clocking

We have a number of processor slots in a row. The processors are each dedicated to a single task that they can complete before going into a busy-looping mode while they wait for the next instruction.

A processor can run at 100% speed if it has enough cooling, and that requires there be no adjacent processor. If a processor has 1 adjacent processor, it only runs at 60% of maximum speed. If it has 2 adjacent processors it can only run at 30% of maximum speed.

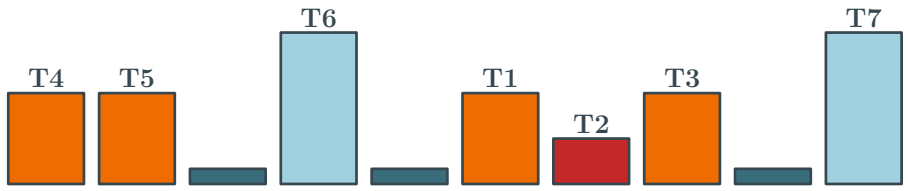


Figure M.1: Illustration of Sample Input 2.

We know how many seconds each task will take running on a processor at full speed, starting from $t = 0$, and need to allocate tasks to processors to ensure the last task finishes as soon as possible.

Input

- The number of tasks to run, n, and the number of processor slots, s ($1 \leq n \leq s \leq 10^6$).
- n further integers giving the duration of each task t_i in seconds if performed at 100% speed ($1 \leq t_i \leq 10^9$).

Output

Output the minimum time by which all of the tasks can be finished.

The output must be accurate to an absolute or relative error of 10^{-6}.

Sample Input 1	Sample Output 1
4 5 1 1000 1 1	1000

Sample Input 2	Sample Output 2
7 10 5 2 3 5 4 6 7	8.33333333

This page is intentionally left (almost) blank.