

## Problem B. Multithreading

Input file: *standard input*  
Output file: *standard output*  
Time limit: 6 seconds  
Memory limit: 512 mebibytes

As improving of single processing unit performance has been stagnating over last decade, many IT companies have to use multiprocessing approach when there is a need in heavy computations.

Yandex is one of such companies, and here, we are constantly inventing state-of-the-art technologies for solving multithreading and multiprocessing problems. As an example of these technologies, we would like to introduce you our new approach on context switch for multithreading tasks.

More formally, we consider  $N$  threads, and for simplicity suppose that each thread is a function which contains  $a_i$  atomic instructions (one atomic instruction takes one time slice to execute).

Each time slice, our planner does the following steps:

1. Consider all threads which have not been executed yet.
2. Pick one of these threads at random with equal probability and execute exactly one of its instructions.  
If all instructions of a thread are executed, the whole thread is considered to be executed.

After all instructions of all threads are executed, we have some order in which the threads became executed. For each thread  $i$ , we are interested in its expected position  $e_i$  in that list.

Formally, the expected position for a thread is calculated as follows. Consider all different orders  $p_1, p_2, \dots, p_{N!}$  in which the threads become executed and the corresponding probabilities  $r_1, r_2, \dots, r_{N!}$  of these orders. The expected place in the list for the thread  $i$  is

$$\sum_{j=1}^{N!} p_j(i) r_j,$$

where  $p_j(i)$  is the position of thread  $i$  in order  $j$ .

### Input

The first line contains the number of threads  $N$  ( $1 \leq N \leq 10^5$ ). The second line contains  $N$  integers  $a_1, a_2, \dots, a_N$  separated by spaces ( $1 \leq a_i \leq 5000$ ).

### Output

Output  $N$  lines:  $i$ -th line must contain the value  $e_i$  with absolute or relative error at most  $10^{-6}$ .

### Example

standard input	standard output
2	1.0000000000
1 5000	2.0000000000