Greek Casino

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.7 seconds
Memory limit:	1024 megabytes

Since the early civilizations, humankind has enjoyed games of chance. Even the ingenious Greeks, known for their groundbreaking concept of the least common multiple (LCM), couldn't resist a good gamble.

Inspired by this mathematical marvel, folks in Athens devised a unique betting system: after purchasing a ticket, a participant would receive a random number of coins. To determine this number, there are $N \geq 3$ ordered slots numbered from 1 to N. A token is initially placed at slot 1, and the following steps are repeated:

- Let x be the number of the slot where the token is currently located.
- Generate a random integer y between 1 and N, and compute z, the LCM of x and y.
- If $z > N$, the procedure ends.
- Otherwise, the token is moved to slot z, and the participant receives one coin.

As it is well known, the house always wins: the casino employs a particular probability distribution for generating random integers, so as to ensure a profitable outcome.

The casino owner is constantly seeking to optimize the betting system's profitability. You, an AI designed to aid in such tasks, are given N and the probability distribution. Determine the expected total number of coins awarded to a participant.

Input

The first line contains an integer N ($3 \leq N \leq 10^5$) indicating the number of slots.

The second line contains N integers $W_1, W_2, \ldots, W_N$ ($1 \leq W_i \leq 1000$ for $i = 1, 2, \ldots, N$), representing that the probability of generating i is $W_i / \left(\sum_j W_j \right)$, that is, the probability of generating i is the relative weight of W_i with respect to the sum of the whole list $W_1, W_2, \ldots, W_N$.

Output

Output a single line with the expected total number of coins awarded to a participant. The output must have an absolute or relative error of at most 10^{-9}. It can be proven that the procedure described in the statement ends within a finite number of iterations with probability 1, and that the expected total number of coins is indeed finite.

Examples

standard input	standard output
3 1 1 1	3.5000000000
3 1 1 2	3.6666666667