Task

There are N bins, numbered $1, 2, \ldots, N$. Initially, for each $1 \leq i \leq N$, the i-th Bin has a_i ($1 \leq a_i \leq 3$) balls.

You will perform the following operation repeatedly until all balls are taken:

Roll a die that shows the numbers $1, 2, \ldots, N$ with equal probabilities, and let i be the outcome. If there are some balls in the i-th Bin, take one of them; if there is none, do nothing.

Determine the expected number of operations needed to take all the balls in $O(N^3)$ time complexity.

Examples

Sample 1

$N = 3$ and $a = \{1, 1, 1\}$. So, all the bins have 1 ball each. Expected number of operations needed to take all the balls is 5.5

The expected number of operations before the first ball is taken, is 1. After that, the expected number of operations before the second ball is taken, is 1.5. After that, the expected number of operations before the third ball is taken, is 3. Thus, the expected number of operations is $1 + 1.5 + 3 = 5.5$.

Sample 2

$N = 1$ and $a = \{3\}$. So, the only bin has 3 balls. Expected number of operations needed to take all the balls is 3.

Sample 3

$N = 2$ and $a = \{1, 2\}$. So, we have two bins; one bin has 1 ball, other one has 2 balls. Expected number of operations needed to take all the balls is 4.5.

Sample 4

$N = 2$ and $a = \{1, 3\}$. Expected number of operations needed to take all the balls is 6.25.

Sample 5

$N = 10$ and $a = \{1, 3, 2, 3, 3, 2, 3, 2, 1, 3\}$. Expected number of operations needed to take all the balls is around 54.4806445749. Note that you need to calculate the exact value up to a relative error of 10^{-9}. Using doubles in Java is enough.