Problem 1: Not Nice Dice
3+1 Points

Problem ID: `diceroll`
Rank: `1+2`

Introduction

You're hosting your first ever slumber party with all your friends, when suddenly you realize you have no games to play! Luckily for you, your good friend Jennifert suggests that everybody play the timeless classic, "Roll the Dice!" The rules are simple: Jennifert rolls a die, and everyone watches you pay her that amount in dollars.

She insists this will turn your failing slumber party around—plus, you're feeling extra confident after trying her "Spell Your Passwords With Your Eyes Closed" challenge last week (surprisingly easy!). You agree to play, but you "innocently" decide to invent a new rule designed to save you money—Jennifert will never notice that you've been cleverly scamming her this whole time!

Problem Statement

Given $\mathbf{N}$ sides of a fair die labeled with the numbers $\mathbf{X_1}$, $\mathbf{X_2}$, … , $\mathbf{X_N}$ on each face, find a number x you can "zero out" to minimize the amount you'll pay Jennifert per dice roll, on average. The faces of the die are **not necessarily unique**. When you roll the die, the number $\mathbf{X_i}$ that appears face-up causes you to pay Jennifert $\mathbf{X_i}$ dollars—unless $\mathbf{X_i}$ equals your chosen x, in which case you don't have to pay her any money (haha, sucker). A die being fair means that each face has an equal probability of appearing face-up when the die is rolled.

*Note: This problem—alongside **all other problems in this contest**—has templates available in Python, Java, and C++! You can find them in the contest.zip provided at the start of the contest. Templates parse the input into a neat function to fill out, so you can jump right into problem solving!*

Input Format

The first line of the input contains a positive integer $\mathbf{T}$ denoting the number of test cases that follow. For each test case:

- The first line contains a single integer $\mathbf{N}$ denoting the number of faces on the die.
- The second line contains $\mathbf{N}$ space-separated integers $\mathbf{X_1}$ $\mathbf{X_2}$ … $\mathbf{X_N}$ denoting the numbers on each face of the die.

Output Format

For each test case, output a number you can “zero out” to minimize the amount you’ll pay Jennifert per dice roll, on average. If there are multiple such values that satisfy this requirement, you may output any.

Problem Constraints

$1 \le \mathbf{T} \le 100$
$1 \le \mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_N} \le 10^3$

Main Test Set

$1 \le \mathbf{N} \le 50$

Bonus Test Set

$1 \le \mathbf{N} \le 10^5$
The sum of $\mathbf{N}$ across all test cases in an input file does not exceed 10^5.

Sample Test Cases

Sample Input *Download*

```
5
7
1 3 5 7 7 7 15
6
3 2 3 2 5 2
8
2 4 4 1 5 16 5 5
4
21 21 21 21
14
12 3 5 5 8 16 8 7 9 10 10 5 8
12
```

Sample Output *Download*

```
7
2
16
21
8
```

Note that this is one of many possible correct outputs. If there are multiple solutions, you may output any of them.

Sample Explanations

Test Case #1:

Zeroing out $x = 7$ gives us a:

- 1/7 chance of paying 1 dollar,
- 1/7 chance of paying 3 dollars,
- 1/7 chance of paying 5 dollars,
- 3/7 chance of paying 0 dollars (by rolling $x = 1$),
- and a 1/7 chance of paying 15 dollars.

This gives us an expected payment of 1/7 + 3/7 + 5/7 + 0 + 15/7 = 24/7 (roughly 3.43) dollars per dice roll, which is the lowest possible amount. Zeroing out $x = 15$, for example, would give us an expected payment of 1/7 + 3/7 + 5/7 + 21/7 + 0 = 30/7 (roughly 4.29) dollars per dice roll.

Test Case #2:

Zeroing out $x = 2$ gives us an expected payment of 11/6 ≈ 1.83 dollars per dice roll, which is the lowest possible amount. Note that zeroing out $x = 3$ also yields this lowest expected value.

Test Case #5:

The three lowest expected payments you can achieve are:

1. 94/14 ≈ 6.71 dollars, achievable by zeroing out $x = 8$ or $x = 12$
2. 98/14 = 7 dollars, achievable by zeroing out $x = 10$

3. 102/14 ≈ 7.29 dollars, achievable by zeroing out $x = 16$