

# Master Chef

**Input File:** *standard input*

**Output File:** *standard output*

Roderick was once the world's greatest light-switch switcher. With astonishing dexterity and lightning speed, Rod's unique capability of hitting many adjacent light-switches simultaneously gave him an edge unparalleled in the light-switching business. However in these tough economic times, Rod has had to turn to alternate pursuits. Yesterday he was called up and offered a place on the popular competitive cooking TV show *Master Chef*, but he needs your help to determine whether he has what it takes to truly master the kitchen.

Roderick will be given  $N$  ingredients lined up in front of him. His strategy for speedy cooking involves selecting, at random, a set of ingredients that form one contiguous subsequence of the line-up (that is, all ingredients between the  $a^{\text{th}}$  and  $b^{\text{th}}$  ingredient inclusive for some random selection of  $a$  and  $b$  such that  $1 \leq a \leq b \leq N$ ). In one swift swoop using one of his exceptionally large hands, Roderick will collect all the selected ingredients and pour them into the pot for mixing. Each ingredient has a tastiness value  $T$  (which may be negative - some ingredients are truly foul), and the tastiness of the final meal will be the *average* tastiness of all selected ingredients. For example, in the following line-up of ingredients, if Roderick happens to select the ones with tastiness 3, -1 and 5, the tastiness of the meal would be  $(3 + -1 + 5)/3 = 2\frac{1}{3}$ .

6	3	-1	5
---	---	----	---

Since Roderick has no idea in advance which subsequence of ingredients he will happen to pick up on the day of the competition, he wants you to write a program that will calculate the expected tastiness of the meal he will create (ie. average tastiness of all possible meals he may create). In the example above with 4 ingredients, there are 10 possible selections of ingredients that Roderick could make meals from. The meal tastiness values would be: 6, 3, -1, 5, 4.5, 1, 2,  $2.\bar{6}$ <sup>1</sup>,  $2.\bar{3}$  and 3.25. The average tastiness of all these 10 possible meals is 2.875. As Roderick isn't great with numbers, he wants you to give the final expected tastiness **rounded to the next integer towards zero** (that is, truncated to zero decimal places). In this example, the answer would hence be 2.

## Input

- The first line of input will contain one integer  $N$ , representing the number of ingredients.
- The following line will contain  $N$  integers, representing the tastiness of each ingredient from left to right.

## Output

Output should consist of a single integer: the (rounded) average tastiness of all possible meals, where a meal's tastiness is the average tastiness of all ingredients in a subsequence of ingredients. Rounding is towards zero.

### Sample Input 1

```
4
6 3 -1 5
```

### Sample Output 1

```
2
```

### Sample Input 2

```
8
-3 10 19 9 7 10 0 5
```

<sup>1</sup> $2.\bar{6}$  means 2.6666666..., or 2.6 recurring

**Sample Output 2**

8

**Sample Input 3**3  
-10 -4 1**Sample Output 3**

-4

**Explanation**

The first sample case corresponds to the example discussed in the problem. In the second sample case, the average tastiness of the 36 possible meals is approximately 8.12, which is truncated to 8. In the third sample case, the average is  $-4.30556$ , which is rounded towards zero to become  $-4$ .

**Constraints & Subtasks**

In all subtasks, the tastiness of each ingredient will be an integer between  $-1000$  and  $1000$  inclusive.

- Subtask 1 (20 points):  $1 \leq N \leq 200$
- Subtask 2 (20 points):  $1 \leq N \leq 2,000$
- Subtask 3 (60 points):  $1 \leq N \leq 200,000$

You will score the full allocated marks for a subtask if your program returns the correct answer for every test case within that subtask. If your program fails any test case in a subtask, your score will be 0 for that subtask.