

You've been enjoying your time at computer school, and you're now taking an interesting course on quantum encryption.

Your instructor has a marking philosophy that's very different from [the last course you took](#) - she believes that every grade should influence your final mark, so your performance will be judged according to the [mean](#) of all your test and assignment grades.

Since you'd like to know how you're doing as the course progresses, you'd like to write an algorithm that calculates your mean grade every time you enter a new mark (ie: every time a graded assignment or test comes back).

Given `scores`, an array of integers representing all your test and assignment grades, your task is to return an array of integers where `output[i]` represents the mean of all your scores up to index `i`. **Your instructor is a harsh marker, so she always rounds the result down to the nearest integer.**

Example

- For `scores = [100, 20, 50, 70, 45]`, the output should be `meanScores(scores) = [100, 60, 56, 60, 57]`.

After each score is entered, the mean is recalculated as follows:

- For `[100]`, the mean is `100` since it's the only element.
- For `[100, 20]`, the mean is $(100 + 20)/2 = 60$.
- For `[100, 20, 50]`, the mean is $(100 + 20 + 50)/3 = 56.667$, which rounds down to `56`.
- For `[100, 20, 50, 70]`, the mean is $(100 + 20 + 50 + 70)/4 = 60$.
- For `[100, 20, 50, 70, 45]`, the mean is $(100 + 20 + 50 + 70 + 45)/5 = 57$.

Input / Output

- **[execution time limit] 0.5 seconds (cpp)**
- **[input] array.integer scores**

An array of integers representing all of your test and assignment grades from the course, in chronological order.

Guaranteed constraints:

$1 \leq \text{scores.length} \leq 5 \cdot 10^4$
 $0 \leq \text{scores}[i] \leq 10^6$

- **[output] array.integer**
 - An array of integers representing the mean grade, rounded down (`output[i]` is the mean of all grades up to index `i` in `scores`).