

## A. Equalize Prices Again

time limit per test: 1 second  
 memory limit per test: 256 megabytes  
 input: standard input  
 output: standard output

You are both a shop keeper and a shop assistant at a small nearby shop. You have  $n$  goods, the  $i$ -th good costs  $a_i$  coins.

You got tired of remembering the price of each product when customers ask for it, thus you decided to simplify your life. More precisely you decided to set the same price for all  $n$  goods you have.

However, you don't want to lose any money so you want to choose the price in such a way that the sum of new prices is not less than the sum of the initial prices. It means that if you sell all  $n$  goods for the new price, you will receive at least the same (or greater) amount of money as if you sell them for their initial prices.

On the other hand, you don't want to lose customers because of big prices so among all prices you can choose you need to choose the minimum one.

So you need to find the minimum possible equal price of all  $n$  goods so if you sell them for this price, you will receive at least the same (or greater) amount of money as if you sell them for their initial prices.

You have to answer  $q$  independent queries.

### Input

The first line of the input contains one integer  $q$  ( $1 \leq q \leq 100$ ) — the number of queries. Then  $q$  queries follow.

The first line of the query contains one integer  $n$  ( $1 \leq n \leq 100$ ) — the number of goods.

The second line of the query contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^7$ ), where  $a_i$  is the price of the  $i$ -th good.

### Output

For each query, print the answer for it — the minimum possible equal price of all  $n$  goods so if you sell them for this price, you will receive at least the same (or greater) amount of money as if you sell them for their initial prices.

### Example

| input     | Copy |
|-----------|------|
| 3         |      |
| 5         |      |
| 1 2 3 4 5 |      |
| 3         |      |
| 1 2 2     |      |
| 4         |      |
| 1 1 1 1   |      |
| output    | Copy |
| 3         |      |
| 2         |      |
| 1         |      |

### Codeforces Round #590 (Div. 3)

[Finished](#)
[Practice](#)


### → Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

[Start virtual contest](#)

### → Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

### → Clone Contest to Mashup

You can clone this contest to a mashup.

[Clone Contest](#)

### → Submit?

 Language: GNU G++11 5.1.0





 Choose file: [选择文件](#) [未选择任何文件](#)
[Submit](#)

### → Problem tags

[math](#) [\\*900](#)

No tag edit access

### → Contest materials

- [Announcement #1 \(en\)](#) 
- [Announcement #2 \(ru\)](#) 
- [Tutorial #1 \(en\)](#) 
- [Tutorial #2 \(ru\)](#) 

[Codeforces](#) (c) Copyright 2010-2019 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Oct/06/2019 10:19:33<sup>UTC+8</sup> (h2).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#)

Supported by



ITMO UNIVERSITY