A. Arithmetic Array

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

An array b of length k is called good if its arithmetic mean is equal to 1. More formally, if $$\frac{b_1 + \cdots + b_k}{k}=1.$$

Note that the value $\frac{b_1+\cdots+b_k}{k}$ is not rounded up or down. For example, the array $[1,1,1,2]$ has an arithmetic mean of 1.25, which is not equal to 1.

You are given an integer array a of length n. In an operation, you can append a **non-negative** integer to the end of the array. What's the minimum number of operations required to make the array good?

We have a proof that it is always possible with finitely many operations.

Input

The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases. Then t test cases follow.

The first line of each test case contains a single integer n ($1 \leq n \leq 50$) — the length of the initial array a.

The second line of each test case contains n integers $a_1,\ldots,a_n$ ($-10^4\leq a_i \leq 10^4$), the elements of the array.

Output

For each test case, output a single integer — the minimum number of non-negative integers you have to append to the array so that the arithmetic mean of the array will be exactly 1.

Example

input

```
4
3
1 1 1
2
1 2
4
8 4 6 2
1
-2
```

output

```
0
1
16
1
```

Note

In the first test case, we don't need to add any element because the arithmetic mean of the array is already 1, so the answer is 0.

In the second test case, the arithmetic mean is not 1 initially so we need to add at least one more number. If we add 0 then the arithmetic mean of the whole array becomes 1, so the answer is 1.

In the third test case, the minimum number of elements that need to be added is 16 since only non-negative integers can be added.

In the fourth test case, we can add a single integer 4. The arithmetic mean becomes $\frac{-2+4}{2}$ which is equal to 1.