

D. Array and GCD

time limit per test: 2 seconds
 memory limit per test: 512 megabytes

You are given an integer array a of size n .

You can perform the following operations any number of times (possibly, zero):

- pay one coin and increase any element of the array by 1 (you must have at least 1 coin to perform this operation);
- gain one coin and decrease any element of the array by 1.

Let's say that an array is *ideal* if both of the following conditions hold:

- each element of the array is at least 2;
- for each pair of indices i and j ($1 \leq i, j \leq n$; $i \neq j$) the greatest common divisor (GCD) of a_i and a_j is equal to 1. If the array has less than 2 elements, this condition is automatically satisfied.

Let's say that an array is *beautiful* if it can be transformed into an ideal array using the aforementioned operations, provided that you initially have no coins. If the array is already ideal, then it is also beautiful.

The given array is not necessarily beautiful or ideal. You can remove any elements from it (including removing the entire array or not removing anything at all). Your task is to calculate the minimum number of elements you have to remove (possibly, zero) from the array a to make it **beautiful**.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 4 \cdot 10^5$).

The second line contains n integers a_1, a_2, \dots, a_n ($2 \leq a_i \leq 10^9$).

Additional constraint on the input: the sum of n over all test cases doesn't exceed $4 \cdot 10^5$.

Output

For each test case, print a single integer — the minimum number of elements you have to remove (possibly, zero) from the array a to make it **beautiful**.

Example

input	Copy
5	
3	
5 5 5	
4	
2 3 2 4	
1	
3	
3	
2 100 2	
5	
2 4 2 11 2	
output	Copy
0	
2	
0	
0	
1	

Note

In the first example, you don't need to delete any elements, because the array is already beautiful. It can be transformed into an ideal array as follows:

Educational Codeforces Round 178 (Rated for Div. 2)

Contest is running

00:29:32

Contestant



→ Submit?

Language: GNU G++23 14.2 (64 bit, ms) ▼

Choose file: No file chosen

→ Last submissions

Submission	Time	Verdict
317638875	Apr/28/2025 19:02	Accepted

$[5, 5, 5] \rightarrow [4, 5, 5] \rightarrow [4, 4, 5] \rightarrow [4, 3, 5]$ (you end up with 3 coins).

In the second example, you need to remove 2 elements so that the array becomes beautiful. If you leave the elements $[2, 3]$ and delete the other elements, then the given array is already ideal (and therefore, beautiful).

In the third example, you don't need to delete any elements, because the array is already ideal (and thus, beautiful).

In the fourth example, the array is beautiful. It can be transformed into an ideal array as follows:
 $[2, 100, 2] \rightarrow [2, 99, 2] \rightarrow [2, 99, 3] \rightarrow [2, 98, 3] \rightarrow [2, 97, 3]$ (you end up with 2 coins).

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Apr/28/2025 18:03:05^{UTC+2} (k1).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



ITMO