

## C. Need More Arrays

time limit per test: 2 seconds  
 memory limit per test: 256 megabytes

Given an array  $a$  and  $n$  integers. It is sorted in non-decreasing order, that is,  $a_i \leq a_{i+1}$  for all  $1 \leq i < n$ .

You can remove any number of elements from the array (including the option of not removing any at all) without changing the order of the remaining elements. After the removals, the following will occur:

- $a_1$  is written to a new array;
- if  $a_1 + 1 < a_2$ , then  $a_2$  is written to a new array; otherwise,  $a_2$  is written to the same array as  $a_1$ ;
- if  $a_2 + 1 < a_3$ , then  $a_3$  is written to a new array; otherwise,  $a_3$  is written to the same array as  $a_2$ ;
- ...

For example, if  $a = [1, 2, 4, 6]$ , then:

- $a_1 = 1$  is written to the new array, resulting in arrays:  $[1]$ ;
- $a_1 + 1 = 2$ , so  $a_2 = 2$  is added to the existing array, resulting in arrays:  $[1, 2]$ ;
- $a_2 + 1 = 3$ , so  $a_3 = 4$  is written to a new array, resulting in arrays:  $[1, 2]$  and  $[4]$ ;
- $a_3 + 1 = 5$ , so  $a_4 = 6$  is written to a new array, resulting in arrays:  $[1, 2]$ ,  $[4]$ , and  $[6]$ .

Your task is to remove elements in such a way that the described algorithm creates as many arrays as possible. If you remove all elements from the array, no new arrays will be created.

### Input

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

The first line of each test case contains one integer  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — the length of the array.

The second line of each test case contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6$ ,  $a_i \leq a_{i+1}$ ) — the elements of the array.

It is guaranteed that the sum of  $n$  across all test cases does not exceed  $2 \cdot 10^5$ .

### Output

For each test case, output one integer — the maximum number of arrays that can be obtained by removing any (possibly zero) number of elements.

### Example

input	Copy
6	
6	
1 2 3 4 5 6	
3	
1 2 3	
4	
1 2 2 4	
1	
2	
3	
1 4 8	
2	
1 1	
output	Copy
3	
2	
2	
1	

### Codeforces Round 1027 (Div. 3)

Contest is running

01:19:44

Contestant



### → Submit?

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

Choose file:  No file chosen

### → Last submissions

Submission	Time	Verdict
<a href="#">321468729</a>	May/26/2025 18:28	Accepted

3  
1

### Note

In the first example, you can remove  $a_3$  and  $a_5$ , then  $a = [1, 2, 4, 6]$ , the process of forming arrays for it is shown in the statement.

In the second example, you need to remove  $a_2$ , after which  $a = [1, 3]$ , and the arrays  $[1]$  and  $[3]$  will be written.

In the third example, no removals are needed; for  $a = [1, 2, 2, 4]$ , the arrays  $[1, 2, 2]$  and  $[4]$  will be written.

---

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: May/26/2025 22:28:54<sup>UTC+7</sup> (k1).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



**ITMO**