2nd Programming Contest League 2025 - University of Birjand

Date: May 2025

Committee Members

Scientific Team
Technical Team
Executive Team

Problem A : Dungeon Equilibrium

An array is called balanced if every integer x that occurs at least once, occurs exactly x times in the array. For example, $[1, 4, 2, 4, 4, 4, 2]$ is balanced, but $[2]$ and $[2, 2, 2]$ are not.

You are given an array a of n elements, $[a_1, a_2, \ldots, a_n]$. The array may not be balanced currently, and you can delete some elements to make it balanced. What is the minimum number of elements you need to delete to make the array balanced?

Input

Each test contains multiple test cases. The first line contains the number of test cases t. The description of test cases follows.

The first line of each test case contains an integer n — the size of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ — the elements of the array a.

Note that there are no constraints on the sum of n over all test cases.

$$1 \leq t \leq 500$$

$$1 \leq n \leq 100$$

$$0 \leq a_i \leq n$$

Output

For each test case, output a single line containing an integer: the minimum number of elements you should remove from the array to make it balanced.

Example

Standard Input	Standard Output
4 3 1 2 2 5 1 1 2 2 3 10 1 2 3 2 4 4 4 4 5 2 1 0	0 2 3 1

Note

In the first test case, the given array is already balanced.

In the second test case, we can delete one occurrence of 1 and one occurrence of 3 to get the array $[1, 2, 2]$ which is balanced.

Problem B : Lantern Festival

Every year, the ancient city of Arvand hosts a magnificent Lantern Festival. Thousands of glowing lanterns rise into the night sky, forming a river of light that stretches far beyond the horizon. For centuries, this tradition has symbolized hope, unity, and the arrival of a prosperous new year.

This year, the mayor of Arvand wants the festival to be brighter than ever. During the opening ceremony, a large group of volunteers marches through the city carrying illuminated lanterns. The visual effect is only successful if strictly more than half of all volunteers are holding lanterns.

The volunteers are divided into three groups:

- **Core Volunteers** — They always attend and always carry lanterns.
- **Opposition Volunteers** — They always attend but this year refuse to carry lanterns.
- **Neutral Volunteers** — They attend, but they have not yet decided whether to participate actively.

The mayor can persuade some Neutral Volunteers to carry lanterns. Your task is to determine the minimum number of Neutral Volunteers that must be convinced so that the total number of lantern-holders becomes strictly greater than half of all volunteers.

If achieving this is impossible (even if all Neutral Volunteers carry lanterns), print `-1`.

Input

A single line containing three space-separated integers:

$$N\ C\ O$$

Where:

- N — total number of volunteers,
- C — number of Core Volunteers (always carry lanterns),
- O — number of Opposition Volunteers (never carry lanterns).

$$1 \le N \le 10^9$$
$$0 \le C, O \le N$$
$$C + O \le N$$

Output

Print a single integer:

- The minimum number of Neutral Volunteers that must be persuaded,
- `-1` if it is impossible.

Example

Standard Input	Standard Output
10 4 3	2

Standard Input	Standard Output
8 3 5	-1

Standard Input	Standard Output
9 5 2	0

Problem C : String LD

StringLD (left delete) is a function that gets a string and deletes its leftmost character (for instance Stringld(”acm”) returns ”cm”).

You are given a list of distinct words, and at each step, we apply stringLD on every word in the list. Write a program that determines the number of steps that can be applied until at least one of the conditions become true:

1) A word becomes empty string, or

2) a duplicate word is generated.

For example, having the list of words aab, abac, and caac, applying the function on the input for the first time results in ab, bac, and aac. For the second time, we get b, ac, and ac. Since in the second step, we have two ac strings, the condition 2 is true, and the output of your program should be 1. Note that we do not count the last step that has resulted in duplicate string. More examples are found in the sample input and output section.

Input

There are multiple test cases in the input. The first line of each test case is n ($1 \leq n \leq 100$), the number of words.

Each of the next n lines contains a string of at most 100 lower case characters.

The input terminates with a line containing 0 (zero).

Output

For each test case, write a single line containing the maximum number of stringLD we can call.

Sample Input and Output

Standard Input	Standard Output
4 aaba aaca baabcd dcba 3 aaa bbbb ccccc 0	1 2

Problem D : Archival System

The university's new archival system requires every stored document identifier (ID) to be a palindrome — a string that reads the same forward and backward (e.g., "level", "abcba"). This rule improves both lookup reliability and system integrity.

However, many old identifiers are not palindromes. The system's administrators have decided that existing characters cannot be removed or replaced; instead, only insertions are allowed. You may insert any lowercase English letter at any position in the string.

Your task is to determine the minimum number of character insertions required to transform a given string into a valid palindrome.

You do not need to construct the resulting palindrome, only compute the minimal number of insertions.

Formally, given a string S consisting of lowercase English letters, compute the minimum number of insertions needed to make S a palindrome.

Example: For input `"race"`, the minimal number of insertions is 3, since `"ecarace"` is one valid optimal palindrome obtained using exactly 3 insertions.

Input

The input contains a single line: A string S — consisting only of lowercase English letters (a–z).

$$1 \leq |S| \leq 3000$$

Output

Print a single integer — the minimum number of insertions needed to make S a palindrome.

Sample Input and Output

Standard Input	Standard Output
`race`	`3`

Explanation: "race" can become "ecarace" using 3 insertions.

Standard Input	Standard Output
`level`	`0`

Explanation: "level" is already a palindrome, so no insertions are needed.

Standard Input	Standard Output
`abca`	`1`

Explanation: inserting one "b" gives the palindrome "abcba".