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".