Palindromic Substrings Detection

Context

In the realms of cybersecurity, computational biology, and text mining, identifying palindromic substrings—substrings that read the same forwards and backwards—can reveal critical insights about structure, anomalies, or encryption schemes. For example, certain DNA segments may be palindromic due to their biological functions, while palindromic patterns in data streams could indicate symmetric encryption blocks or repeated logic patterns. Therefore, efficiently finding all palindromic substrings in a given string has both theoretical and practical significance.

Problem Statement

You are given a string S composed of lower case english letters, i.e., characters with ASCII values between 97 and 122 inclusive. Your task is to compute the number of distinct substrings of S that are palindromic. A substring is defined as any contiguous sequence of characters from S, and a palindrome is a substring that reads the same forwards and backwards.

Input

The input consists of a single line containing the string S, with the following constraints:

- $1 \leq |S| \leq 1{,}000$
- Each character in S has an ASCII value between 97 and 122 (inclusive), i.e., lower case letters.

Output

Print a single integer: the number of non-empty substrings of S that are palindromes.

Clarifications

- Substrings that are identical but appear at different positions in the string are counted multiple times. For example, in `aaa`, the substring `a` appears 3 times and should be counted 3 times.

Example

Input 1:

```
ababa
```

Output 1:

```
9
```

Input 2:

```
abracadabra
```

Output 2:

```
13
```