**[Palindromic Substrings](https://leetcode.com/problems/palindromic-substrings/)**

Given a string s, return *the number of****palindromic substrings****in it*.

A string is a **palindrome** when it reads the same backward as forward.

A **substring** is a contiguous sequence of characters within the string.

**Example 1:**

**Input:** s = "abc"

**Output:** 3

**Explanation:** Three palindromic strings: "a", "b", "c".

**Example 2:**

**Input:** s = "aaa"

**Output:** 6

**Explanation:** Six palindromic strings: "a", "a", "a", "aa", "aa", "aaa".

**Constraints:**

* 1 <= s.length <= 1000
* s consists of lowercase English letters.

CODE

class Solution {

public:

    int countSubstrings(string s) {

        int n = s.length(), ans = 0;

        for (int i = 0; i < n; ++i) {

            int even = palindromeCount(s, i, i + 1);

            int odd = palindromeCount(s, i, i);

            ans += even + odd;

        }

        return ans;

    }

    int palindromeCount(const string& s, int left, int right) {

        int count = 0;

        while (left >= 0 && right < s.length() && s[left] == s[right]) {

            --left;

            ++right;

            ++count;

        }

        return count;

    }

};

Link : <https://leetcode.com/problems/palindromic-substrings/?envType=daily-question&envId=2024-02-10>