# Special Palindrome Again ☆

Problem Submissions Leaderboard Editorial

A string is said to be a special palindromic string if either of two conditions is met:

- All of the characters are the same, e.g. aaa.
- All characters except the middle one are the same, e.g. aadaa.

A special palindromic substring is any substring of a string which meets one of those criteria. Given a string, determine how many special palindromic substrings can be formed from it.

For example, given the string $s =$ **mnonopoo**, we have the following special palindromic substrings:
**{m, n, o, n, o, p, o, o, non, ono, opo, oo}**.

**Function Description**

Complete the substrCount function in the editor below. It should return an integer representing the number of special palindromic substrings that can be formed from the given string.

substrCount has the following parameter(s):

- n: an integer, the length of string s
- s: a string

**Input Format**

The first line contains an integer, $n$, the length of $s$.
The second line contains the string $s$.

**Constraints**

$1 \leq n \leq 10^6$
Each character of the string is a lowercase alphabet, **ascii[a-z]**.

**Output Format**

Print a single line containing the count of total special palindromic substrings.

**Sample Input 0**

```
5
asasd
```

**Sample Output 0**

```
7
```

**Explanation 0**

The special palindromic substrings of $s =$ **asasd** are **{a, s, a, s, d, asa, sas}**

**Sample Input 1**

```
7
abcbaba
```

**Sample Output 1**

```
10
```

**Explanation 1**

The special palindromic substrings of $s =$ **abcbaba** are **{a, b, c, b, a, b, a, bcb, bab, aba}**

**Sample Input 2**

```
4
aaaa
```

**Sample Output 2**

```
10
```

**Explanation 2**

The special palindromic substrings of $s =$ **aaaa** are **{a, a, a, a, aa, aa, aa, aaa, aaa, aaaa}**

C#

```
14
15  class Solution {
16
17
18      // Complete the substrCount function below.
19      static long substrCount(int n, string s) {
20
21          return countPalindrome(s, 0, s.Length - 1) + s.Length;
22
23      }
24
25      static long countPalindrome(string s, int start, int end)
26  {
27      if (start > end)
28          return 0;
29
30      long count = 0;
31      int mid = (end + start) / 2;
```