Special String Again ☆

Your Special String Again submission got 40.00 points. Share Tweet

Try the next challenge

Problem Submissions Leaderboard Editorial

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

For example, given the string $s =$ **mnonopoo**, we have the following special 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 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 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 $s = \texttt{abcbaba}$ are $\{\texttt{a, b, c, b, a, b, a, bcb, bab, aba}\}$

Sample Input 2

```
4
aaaa
```

Sample Output 2

```
10
```

Explanation 2

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

Change Theme C++

```
6  long substrCount(int n, string s) {
7      long count = n;
8      int p1, p2;
9      for (int i =0; i< n; i++){
10         p1=0;
11         p2=1;
12         while((s[i] == s[i+1]) && (i+1)<n){
13             p1++;
14             i++;
15         }
16         count += (p1*(p1+1))/2;
17
18         while((s[i-p2] == s[i-1] && s[i-1] ==s[i+p2]) && ((i-p2)>=0 && (i+p2)<n)){
19             count++;
20             p2++;
21         }
22     }
23
24 return count;
25
26 }
```

Line: 11 Col: 14

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

```
Success
```

Input (stdin) Download

```
1 5
2 asasd{-truncated-}
```

Download to view the full testcase

Expected Output Download

```
1 7
```