**Substrings with similar first and last characters**

Submissions: [1862](https://practice.geeksforgeeks.org/problem_submissions.php?pid=1740)  Accuracy:

46.15%

   Difficulty: [Basic](https://practice.geeksforgeeks.org/Basic/0/0/)   Marks: 1

Show Topic Tags   

Given a string s, find out the count of all contiguous substrings whose starting and ending are same character.  
Note: string contains lowercase English alphabets only.  
  
Example

Input : S = "abcab"

Output : 7

There are 15 substrings of "abcab"

a, ab, abc, abca, abcab, b, bc, bca

bcab, c, ca, cab, a, ab, b

Out of the above substrings, there

are 7 substrings : a, abca, b, bcab,

c, a and b.

Input : S = "aba"

Output : 4

The substrings are a, b, a and aba

**Input:**  
The first line of each test case contains an integer T denoting the number of test cases. Then T test cases follows. The first line of each test case contains a number N denoting the length of the string (S). Then the next line contains the string S.  
  
**Output:**  
For each test case output a new line containing a single integer, denoting the count of all the substrings whose first and last character are same.  
  
**Constraints:**  
1<=T<=100  
1<=N<=104

**Example:  
Input:**  
2  
5  
abcab  
13  
geeksforgeeks  
**Output:**  
7  
22

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/substrings-with-similar-first-and-last-characters/0#ExpectOP) option \*\*

<https://practice.geeksforgeeks.org/problems/substrings-with-similar-first-and-last-characters/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int ObtenerCadenas(string s)

{

Dictionary<char, int> dic =

new Dictionary<char, int>();

for(int i =0; i<s.Length; i++)

{

if(dic.ContainsKey(s[i])) dic[s[i]]++;

else dic[s[i]] = 1;

}

int sum = 0;

foreach(KeyValuePair<char,int> kvp in dic)

{

sum += (kvp.Value \* (kvp.Value + 1)) / 2;

}

return sum;

}

static void Main(string[] args)

{

//string s = "geeksforgeeks";

//Console.WriteLine( ObtenerCadenas(s));

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

int n = int.Parse(Console.ReadLine());

string s = Console.ReadLine().Trim();

Console.WriteLine(ObtenerCadenas(s));

}

Console.ReadLine();

}

}

}