**Substrings with similar first and last characters**

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 Examples Use Expected Output\*\*

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

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication248;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Map;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication248 {

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

String s = br.readLine();

HashMap<Character, Integer> hm = new HashMap<Character, Integer>();

int cont =0;

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

if(hm.containsKey(s.charAt(i))) {

hm.put(s.charAt(i), hm.get(s.charAt(i))+1);

//cont++;

}else{

hm.put(s.charAt(i), 1);

//cont++;

}

}

// System.out.println(cont );

int sum =0;

for (Map.Entry<Character, Integer> entry : hm.entrySet())

{

//System.out.println(entry.getKey() + "/" + entry.getValue());

int val = entry.getValue();

sum += (val\* (val+1))/2;

}

System.out.println(sum);

}

}

}