# Palindrome Pairs

Time Limit 2 seconds
Memory Limit 256 megabytes

After learning a lot about space exploration, a little girl named Ana wants to change the subject.

Ana is a girl who loves palindromes (string that can be read the same backwards as forward). She has learned how to check for a given string whether it's a palindrome or not, but soon she grew tired of this problem, so she came up with a more interesting one and she needs your help to solve it:

You are given an array of strings which consist of only small letters of the alphabet. Your task is to find **how many** palindrome pairs are there in the array. A palindrome pair is a pair of strings such that the following condition holds: **at least one** permutation of the concatenation of the two strings is a palindrome. In other words, if you have two strings, let's say "aab" and "abcac", and you concatenate them into "aababcac", we have to check if there exists a permutation of this new string such that it is a palindrome (in this case there exists the permutation "aabccbaa").

## Input

The first line contains a positive integer N ($1 \le N \le 100000$), representing the length of the input array.
Eacg of the next N lines contains a string (consisting of lowercase English letters from 'a' to 'z') — an element of the input array.
The total number of characters in the input array will be less than 1000000.

## Output

Output one number, representing **how many palindrome pairs** there are in the array.

| Sample Input 1 | Sample Output 1 |
| --- | --- |
| 3<br>aa<br>bb<br>cd | 1 |

| Sample Input 2 | Sample Output 2 |
| --- | --- |
| **6**<br>**aab**<br>**abcac**<br>**dffe**<br>**ed**<br>**aa**<br>**aade** | **6** |

# Note

The first example:

1. aa ++ bb ⟶⟶ abba.

The second example:

1. aab ++ abcac == aababcac ⟶⟶ aabccbaa
2. aab ++ aa == aabaa
3. abcac ++ aa == abcacaa ⟶⟶ aacbcaa
4. dffe ++ ed == dffeed ⟶⟶ fdeedf
5. dffe ++ aade == dffeaade ⟶⟶ adfaafde
6. ed ++ aade == edaade ⟶⟶ aeddea