

Conta le coppie di anagrammi (anagrammi)

Una stringa ottenuta permutando caratteri di una stringa a è detta anagramma di a .

Ad esempio, “greenbin” è un anagramma di “beginner”. Come si può vedere, se uno stesso carattere compare più volte in una stringa a , un anagramma di a contiene quel carattere lo stesso numero di volte.

Ti sono date N stringhe s_1, s_2, \dots, s_N . Ogni stringa è composta esattamente da 10 caratteri e contiene solo lettere minuscole. Inoltre tutte le stringhe sono distinte. Calcola quante sono le coppie di interi i, j ($1 \leq i < j \leq N$) tali che s_i è un anagramma di s_j .

🔔 La risposta potrebbe non essere rappresentabile in un intero a 32-bit, ma è sempre rappresentabile in un intero a 64-bit. In C++/C per rappresentare un intero a 64-bit è possibile usare il tipo di dato `long long`, mentre in Java è possibile usare il tipo di dato `long`. In python 3 invece il tipo di dato `int` è sufficiente.

Dati di input

La prima riga contiene l'intero N . Le successive N righe contengono le stringhe s_1, s_2, \dots, s_N .

Dati di output

Stampa il numero di coppie di interi i, j ($1 \leq i < j \leq N$) tali che s_i è un anagramma di s_j .

Assunzioni

- $2 \leq N \leq 10^5$.
- s_i è una stringa di lunghezza 10 e contiene solo lettere minuscole ($1 \leq i \leq N$).
- s_1, s_2, \dots, s_N sono tutte distinte.

Esempi di input/output

input	output
3 acornistnt peanutbomb constraint	1
2 oneplustwo ninemodsix	0
5 abaaaaaaaa oneplustwo aaaaaaaaaba twoplusone aaaabaaaaa	4

Spiegazione

Nel primo caso d'esempio s_1 ="acornistnt" è un anagramma di s_3 ="constraint". Non ci sono altre coppie i, j valide, quindi la risposta è 1.


Nel secondo caso d'esempio non ci sono coppie i, j tali che s_i è un anagramma di s_j .

Conta le coppie di anagrammi (anagrammi)

We will call a string obtained by arranging the characters contained in a string a in some order, an anagram of a .

For example, “greenbin” is an anagram of “beginner”. As seen here, when the same character occurs multiple times, that character must be used that number of times.

Given are N strings s_1, s_2, \dots, s_N . Each of these strings has a length of 10 and consists of lowercase English characters. Additionally, all of these strings are distinct. Find the number of pairs of integers i, j ($1 \leq i < j \leq N$) such that s_i is an anagram of s_j .

 The answer may not be representable in a 32-bit integer, but it is always representable in a 64-bit integer. In C++/C you can use the `long long` data type to represent a 64-bit integer, while in Java you can use the `long` data type. In python 3 however the data type `int` is enough.

Input

The first line contains the integer N . The following N lines contains s_1, s_2, \dots, s_N .

Output

Print the number of pairs of integers i, j ($1 \leq i < j \leq N$) such that s_i is an anagram of s_j .

Constraints

- $2 \leq N \leq 10^5$.
- s_i is a string of length 10 and consists of lowercase English characters ($1 \leq i \leq N$).
- s_1, s_2, \dots, s_N are all distinct.

Examples

input	output
3 acornistnt peanutbomb constraint	1
2 oneplustwo ninemodsix	0
5 abaaaaaaaa oneplustwo aaaaaaaaaba twoplusone aaaabaaaaa	4

Explanation

In the first sample s_1 ="acornistnt" is an anagram of s_3 ="constraint". There are no other pairs i, j such that s_i is an anagram of s_j , so the answer is 1.

In the second sample there is no pairs i, j such that s_i is an anagram of s_j .