Problem A. 2-Letter Strings

Time limit 2000 ms
Mem limit 262144 kB

Given n strings, each of length 2, consisting of lowercase Latin alphabet letters **from 'a' to 'k'**, output the number of pairs of indices (i, j) such that $i < j$ and the i-th string and the j-th string differ in exactly one position.

In other words, count the number of pairs (i, j) $(i < j)$ such that the i-th string and the j-th string have **exactly** one position p $(1 \le p \le 2)$ such that $s_{ip} \neq s_{jp}$.

The answer may not fit into 32-bit integer type, so you should use 64-bit integers like `long long` in C++ to avoid integer overflow.

Input

The first line of the input contains a single integer t $(1 \le t \le 100)$ — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 10^5)$ — the number of strings.

Then follows n lines, the i-th of which containing a single string s_i of length 2, consisting of lowercase Latin letters **from 'a' to 'k'**.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, print a single integer — the number of pairs (i, j) $(i < j)$ such that the i-th string and the j-th string have **exactly** one position p $(1 \le p \le 2)$ such that $s_{ip} \neq s_{jp}$.

Please note, that the answer for some test cases won't fit into 32-bit integer type, so you should use at least 64-bit integer type in your programming language (like `long long` for C++).

Sample 1

Input	Output
4 6 ab cb db aa cc ef 7 aa bb cc ac ca bb aa 4 kk kk ab ab 5 jf jf jk jk jk	5 6 0 6

Note

For the first test case the pairs that differ in exactly one position are: ("`ab`", "`cb`"), ("`ab`", "`db`"), ("`ab`", "`aa`"), ("`cb`", "`db`") and ("`cb`", "`cc`").

For the second test case the pairs that differ in exactly one position are: ("`aa`", "`ac`"), ("`aa`", "`ca`"), ("`cc`", "`ac`"), ("`cc`", "`ca`"), ("`ac`", "`aa`") and ("`ca`", "`aa`").

For the third test case, the are no pairs satisfying the conditions.