Task 3: CatsAndHams

The new archotech computers all use high end computing hardware to keep a lookout for the imminent mechanoid swarm. Archotechnology is misunderstood at best even by GlitterWorld standards. Here's a quick rundown.

Instead of traditional bits of 1s and 0s, data is stored as Cats and Hams. This data can be represented as streams containing the characters 'c' and 'h', where each stream contains a total of 32 characters.

These computers are on constant lookout for 2 streams of data that indicate **volatility**. Volatility would indicate that the mechanoid swarm is quickly approaching. It is of utmost importance as this acts as the only alert system available. Volatility is defined as such:

Given a number of data streams, we can split each stream into 8 sections of 4 characters each. Volatility is confirmed when any 2 streams contain all possible permutations of the sections containing 'c' and 'h'.

Currently, it seems the systems just aren't processing the data fast enough. With lives on the line, can you write a script that can do this quickly?

Input format

Your program must read from standard input.

The first line contains an integer N that indicates how many streams of data need to be processed.

The next N lines contain strings of 32 characters each that consist of the characters 'c' and 'h'.

Output format

Your program must print to standard output.

The output consists of a single integer, which is the number of pairs of data streams that are considered 'volatile'.

Constraints

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances that satisfy the following limits:

- $1 \leq N \leq 5 \times 10^5$

Subtask	Marks	Additional Limits
1	20	$1 \leq N \leq 10^3$
2	30	$N \leq 10^4$
3	50	No further restrictions

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
2 ccccccchcchccchhchccchchchhcchhh hhhhhhhchhchhhcchchhhchchcchhccc	1

Explanation

All possible permutations of a section with 4 characters are as such:

cccc	chcc	hhhh	hchh
ccch	chch	hhhc	hchc
cchc	chhc	hhch	hcch
cchh	chhh	hhcc	hccc

Comparing the 2 data streams, all permutations can be found when splitting each stream into 8 sections each. Hence, there is 1 pair of data streams that is volatile.

Sample Testcase 2

This testcase is valid for all subtasks.

Input	Output
3 ccccccchcchccchhchccchchchhcchhh hhhhhhhchhchhhcchchhhchchcchhccc hhhhhhhchhchhhcchchhhchchcchhccc	2

Explanation

All possible permutations of a section with 4 characters are found when considering streams {1, 2} and {1, 3}

Each pair is counted as 1 count of volatility, hence there are 2 pairs of volatile data streams.

Sample Testcase 3

This testcase is valid for all subtasks.

Input	Output
6 ccccccchcchccchhchccchchchhcchhh chccchchchhcchhhccccccchcchccchh hhhhhhhchhchhhcchchhhchchcchhccc hchhhchchcchhccchhhhhhhchhchhhcc cccccccccccccccccccccccccccccccc hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh	4

There are 4 possible pairs of streams here that indicate volatility, {1, 3}, {1, 4}, {2, 3} and {2, 4}