Palindromic Birthday (`birthday`)

William has been invited to Antonio's birthday party, where he knows there will be a lot of drinks, food, and games to socialise with friends. Antonio has a lot of games to play with, but as a good computer scientist, his games are mostly designed to test how good his friends are at problem-solving. Often, in these games, one is required to be able to find a solution to a problem and to be quick in doing it.

For fairness sake, Antonio let William pick today's game and so they all start playing. At the beginning of the game, there is a black box from whom every participant randomly picks a string S_i. Once every one of the N participants has chosen their string, they have to try to pair them up with each other's strings, attempting to create palindromes while doing so.

More specifically, they can choose any S_i and S_j $(i \neq j)$ and attempt to concatenate them to make a palindrome string. The winner of the game is the fastest person who finds the exact number of ways they can create a palindrome string among all possible pairs.

Figure 1: The palindrome-finding game.

Help William win the game by writing an algorithm which, given N unique strings, calculates the number of ways a palindrome string can be constructed!

Input

The first line contains one integer N, the number of players.

Each of the next N lines contains a string S_i, the string chosen by the i-th player.

Output

You need to write a single line containing the number of ways we can create a palindrome string.

Constraints

- $2 \leq N \leq 10\,000$.
- $1 \leq |S_i| \leq 600$, where $|S_i|$ is the string length.
- Each string is unique.
- Each string is always formed by lowercase English letters.

Examples

input	output
5 abcd dcba lls s sssll	4
3 bat tab cat	2

Explanation

In the first example there are 4 ways to create a palindrome string:

- `abcddcba` formed by S_0 and S_1.

- `dcbaabcd` formed by S_1 and S_0.

- `slls` formed by S_3 and S_2.

- `llsssll` formed by S_2 and S_4.