Triangle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given n strings $S_1, S_2, \cdots, S_n$ consisting of lower-cased English letters, we say three strings S_a, S_b and S_c form a triangle, if all the following constraints are satisfied:

- $S_a + S_b > S_c$ or $S_b + S_a > S_c$.
- $S_a + S_c > S_b$ or $S_c + S_a > S_b$.
- $S_b + S_c > S_a$ or $S_c + S_b > S_a$.

Here $+$ is the string concatenation operation and strings are compared by lexicographic order. For example, `ba`, `cb` and `cbaa` forms a triangle, because:

- `cb` + `ba` = `cbba` > `cbaa`.
- `cbaa` + `ba` = `cbaaba` > `cb`.
- `cb` + `cbaa` = `cbcbaa` > `ba`.

Count the number of integer tuples (a, b, c) such that $1 \le a < b < c \le n$ and S_a, S_b, S_c forms a triangle.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 3 \times 10^5$) indicating the number of strings.

For the following n lines, the i-th line contains a string S_i ($1 \le |S_i| \le 3 \times 10^5$) consisting of lower-cased English letters.

It's guaranteed that the total length of the strings in a single test case does not exceed 3×10^5, and the total length of strings of all test cases does not exceed 10^6.

Output

For each test case, output one line containing one integer indicating the number of valid tuples.

Example

standard input	standard output
3 6 cbaa cb cb cbaa ba ba 3 sdcpc sd cpc 1 ccpc	16 0 0