

Отчетност

Бобо записва най-старателно кои овчици влизат в помещението за доене. На всеки час три овчици влизат за доене и Бобо записва имената им. Например за един 5-часов период може да се получи следния списък, където всеки ред съответства на една тройка овчици, влизали за доене:

BESSIE ELSIE MATILDA

FRAN BESSIE INGRID

BESSIE ELSIE MATILDA

MATILDA INGRID FRAN

ELSIE BESSIE MATILDA

Бобо забелязал, че една и съща група овчици може да се появява няколко пъти в неговия списък, макар и имената им да не са записани в същия ред. В примера тройката овчици BESSIE, ELSIE и MATILDA се появява три пъти. Помогнете на Бобо да намери броя на срещанията на групата овчици, която е влизала най-много пъти за доене.

Жокер: Тривиалната идея е да опитате всички възможности: проверявате колко пъти всяка група се появява в списъка. Знаем, че редът на овцете може да бъде различен в едни и същи групи. Изпробването на всички пермутации на овце в група усложнява кодирането. Вместо това, по-добра идея е да сортирате овцете по азбучен ред в група и да ги съхранявате в низ. След това можем лесно да търсим всеки низ в списъка. Например, извадката ще бъде въведена по следния начин след сортиране на всяка група:

BESSIE ELSIE MATILDA

BESSIE FRAN INGRID

BESSIE ELSIE MATILDA

FRAN INGRID MATILDA

BESSIE ELSIE MATILDA

Сега можем ясно да видим, че низът "BESSIE ELSIE MATILDA" се появява в списъка три пъти. Обърнете внимание, че поставяме разстояние между имената на овцете.

След това сортирайте списъка и сигурно вече сами се досещате как с линейна сложност да преброите всяка уникална група, колко пъти е влизала за доене след като списъка ви изглежда така.

BESSIE ELSIE MATILDA

BESSIE ELSIE MATILDA

BESSIE ELSIE MATILDA

BESSIE FRAN INGRID

FRAN INGRID MATILDA

Второ решение (Оптимално по скорост): Не сортирайте списъка - това изисква време ($O(N * \log N)$). Използвайте `std::unordered_map` (`#include`), за да преброите всеки стринг колко пъти се среща. `std::unordered_map` е по-бърз от `std::map`, защото използва хеш-таблица във вътрешната си реализация, вместо червено-черно балансирано (сортирано) двоично дърво - реализацията на `std::map`. Използването на `std::unordered_map` в тази задача е оправдано, тъй като нейното решение не зависи от това дали стринговете в мапа (`unordered_map`) са подредени вътрешно по азбучен ред.

Input Format

За всеки тест на първия ред са зададени броят на часовете, N , за които Бобо е водел записки ($1 \leq N \leq 1000$). Следват N реда, всеки съдържащ имената на три овчици, разделени с интервали. Всяко име се състои от 1 до 10 букви, като са използвани само буквите A-Z. Края на входа е маркиран с нула.

Constraints

$1 \leq N \leq 1000$

Output Format

За всеки тест на отделен ред на стандартния изход извеждайте броят на срещанията на групата, която е влизала за доене най-много пъти.

Sample Input 0

```
5
BESSIE ELSIE MATILDA
FRAN BESSIE INGRID
BESSIE ELSIE MATILDA
MATILDA INGRID FRAN
ELSIE BESSIE MATILDA
0
```

Sample Output 0

```
3
```