

Team Naming

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 512 megabytes

Naming is always difficult, and coming up with a name for a team competing in XCPC is even harder.

Kagarii has been racking his brain over how to come up with a unique, distinctive, grand, profound, and cleverly crafted team name that resonates with all three members of the team. It is even more challenging to satisfy the tastes of all three members.

In the end, desperate Kagarii thought of a very trite method: to select one character from each team member's name to form a three-character team name. However, he suddenly realized that the characters selected from the names of the three members happen to form the name of one of the team members!

For example, in Kagarii's team, the names of the three members are: "Da Da Juan", "Xiao Zhong Da", and "Da Zhong Xiao". These three can form the team name "Da Zhong Xiao" (the first character of "Da Da Juan", the second character of "Xiao Zhong Da", and the third character of "Da Zhong Xiao"), which happens to be the name of one of the members.

Clearly, this makes naming the team much easier, so Kagarii plans to promote this naming method to the entire training team. He hopes to see more such team names.

Specifically, the training team has n members, where the name of the i -th member consists of three numbers $S_{i,1}, S_{i,2}, S_{i,3}$ (since there are too many Chinese characters, they are all encoded as integers), and it is guaranteed that all members have different names. Kagarii wants to know how many ways there are to select a team of three members such that selecting one character from each person's name at **different positions**, and combining these characters according to the positions of the characters in the **original names**, exactly forms the name of one of the team members. Of course, if the formed team names are different, they are considered different solutions.

Formally, the task is to count the number of quadruples (i, j, k, id) , where $1 \leq i < j < k \leq n$ and $id \in \{i, j, k\}$, such that there exists a permutation p of $\{i, j, k\}$ satisfying $\forall x \in \{1, 2, 3\}, S_{id,x} = S_{p_x,x}$.

Input

The first line contains a positive integer n ($3 \leq n \leq 10^5$), representing the number of members.

The next n lines each contain three positive integers $S_{i,1}, S_{i,2}, S_{i,3}$ ($1 \leq S_{i,j} \leq 10^6$), representing each person's name. It is guaranteed that for any $i \neq j$, there exists an x such that $S_{i,x} \neq S_{j,x}$.

Output

Output a single line containing an integer, representing the number of valid quadruples.

Examples

standard input	standard output
5 4 2 4 2 4 3 2 1 2 3 4 4 4 4 1	7
3 1 2 3 1 2 4 2 2 3	3

Note

For the sample 1, a quadruple satisfying the condition is $(2, 3, 5, 2)$, and the corresponding permutation p is $\{3, 5, 2\}$.

For the sample 2, quadruples satisfying the condition are $(1, 2, 3, 1)$, $(1, 2, 3, 2)$, $(1, 2, 3, 3)$.