



4-2. FunctionCup Museum

To celebrate the 4th FunctionCup, we made a museum exhibiting the history of FunctionCup. In the museum, there are several booths where visitors can take a look at the past problems of FunctionCup. When the tour is over, each visitor takes souvenirs from the museum - one name tag, one T-shirt, and one gifticon.

- *Name tag*: Champions of each task in the first FunctionCup received this prize.
 - Name tags for different tasks had different colors.
 - There are 100 types of name tags numbered 1 through 100.
- *T-shirt*: Champions of each task in the second FunctionCup received this prize.
 - T-shirts for different tasks had different designs.
 - There are 100 types of T-shirts numbered 1 through 100.
- *Gifticon*: Champions of each task in the third FunctionCup received this prize. (translator's note: *gifticon* is a Korean brand, you can think this as online gift cards)
 - Gifticons for different tasks were different.
 - There are 100 types of gifticons numbered 1 through 100.

Note that the museum has an infinite amount of souvenirs so that all visitors can always take souvenirs of types they prefer.

The curator of the museum is trying to group visitors by their preferences of souvenirs. The curator decides to make a graph where each vertex corresponds to each visitor, and two visitors are directly connected by an edge if and only if they have similar preferences on souvenirs.

Let's say two distinct visitors have *similar preferences* if there exists at least one common souvenir that both visitors took from the museum.

You are given the types of souvenirs that each visitor has taken from the museum. Write a program that computes the number of pairs of visitors that have *similar preferences*.

Implementation details

You should implement the following function. It will be called by the grader once for each test case.

```
int64 CountSimilarPairs(int[] B, int[] T, int[] G)
```

- B, T, G : integer arrays of length N . For each i ($0 \leq i \leq N - 1$), the visitor i brought name tag $B[i]$, T-shirt $T[i]$ and gifticon $G[i]$.
- The function should return the number of pairs of visitors that have similar preferences.

Example

Consider the following call:

```
CountSimilarPairs([1, 2, 3, 4, 1], [1, 2, 1, 2, 1], [1, 1, 2, 2, 3])
```

The pairs with similar preferences are (0, 1), (0, 2), (1, 3), (2, 3), (0, 4), and (2, 4).

The answer is 6.

Constraints

- $1 \leq N \leq 200\,000$
- $1 \leq B[i], T[i], G[i] \leq 100$ (for each $0 \leq i \leq N - 1$)

Subtasks

1. (27 points) $1 \leq N \leq 2\,000$
2. (73 points) No additional constraints.

Sample grader

You can download the sample grader package on the same page you downloaded the problem statement. (scroll down if you don't see the attachment)

If you use IDEs like Visual Studio, Eclipse or Code::Blocks, then import `museum.cpp`, `museum.h` and `grader.cpp` into one project and you will be able to compile all these files at once.

If you want to compile by yourself, refer to the compilation commands in the statement page.

You should submit only `museum.cpp`.

Input format

- line 1: N
- line 2: $B[0] B[1] \dots B[N - 1]$
- line 3: $T[0] T[1] \dots T[N - 1]$
- line 4: $G[0] G[1] \dots G[N - 1]$

Output format

The sample grader prints a single line containing the return value of `CountSimilarPairs`.