



## B. New Year and Ascent Sequence

time limit per test: 2 seconds

memory limit per test: 1024 megabytes

A sequence  $a = [a_1, a_2, \dots, a_l]$  of length  $l$  has an **ascent** if there exists a pair of indices  $(i, j)$  such that  $1 \leq i < j \leq l$  and  $a_i < a_j$ . For example, the sequence  $[0, 2, 0, 2, 0]$  has an ascent because of the pair  $(1, 4)$ , but the sequence  $[4, 3, 3, 3, 1]$  doesn't have an ascent.

Let's call a concatenation of sequences  $p$  and  $q$  the sequence that is obtained by writing down sequences  $p$  and  $q$  one right after another without changing the order. For example, the concatenation of the  $[0, 2, 0, 2, 0]$  and  $[4, 3, 3, 3, 1]$  is the sequence  $[0, 2, 0, 2, 0, 4, 3, 3, 3, 1]$ . The concatenation of sequences  $p$  and  $q$  is denoted as  $p + q$ .

Gyeonggeun thinks that sequences with ascents bring luck. Therefore, he wants to make many such sequences for the new year. Gyeonggeun has  $n$  sequences  $s_1, s_2, \dots, s_n$  which may have different lengths.

Gyeonggeun will consider all  $n^2$  pairs of sequences  $s_x$  and  $s_y$  ( $1 \leq x, y \leq n$ ), and will check if its concatenation  $s_x + s_y$  has an ascent. Note that he may select the same sequence twice, and the order of selection matters.

Please count the number of pairs  $(x, y)$  of sequences  $s_1, s_2, \dots, s_n$  whose concatenation  $s_x + s_y$  contains an ascent.

### Input

The first line contains the number  $n$  ( $1 \leq n \leq 100\,000$ ) denoting the number of sequences.

The next  $n$  lines contain the number  $l_i$  ( $1 \leq l_i$ ) denoting the length of  $s_i$ , followed by  $l_i$  integers  $s_{i,1}, s_{i,2}, \dots, s_{i,l_i}$  ( $0 \leq s_{i,j} \leq 10^6$ ) denoting the sequence  $s_i$ .

It is guaranteed that the sum of all  $l_i$  does not exceed 100 000.

### Output

Print a single integer, the number of pairs of sequences whose concatenation has an ascent.

### Examples

#### input

```
5
1 1
1 1
1 2
1 4
1 3
```

[Copy](#)

#### output

```
9
```

[Copy](#)

#### input

```
3
4 2 0 2 0
6 9 9 8 8 7 7
1 6
```

[Copy](#)

#### output

```
7
```

[Copy](#)

#### input

[Copy](#)

### Hello 2020

**Finished**

Practice



### → Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

[Start virtual contest](#)

### → Clone Contest to Mashup

You can clone this contest to a mashup.

[Clone Contest](#)

### → Submit?

Language: [GNU G++17 7.3.0](#)

Choose file:  No file chosen

[Submit](#)

### → Contest materials

- Hello 2020 (en) [X](#)
- Hello 2020 Editorial (en) [X](#)

```
10
3 62 24 39
1 17
1 99
1 60
1 64
1 30
2 79 29
2 20 73
2 85 37
1 100
```

**output**

```
72
```

**Note**

For the first example, the following 9 arrays have an ascent:

[1, 2], [1, 2], [1, 3], [1, 3], [1, 4], [1, 4], [2, 3], [2, 4], [3, 4]. Arrays with the same contents are counted as their occurrences.

---

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Aug/30/2025 14:57:00<sup>UTC-4</sup> (i2).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



| ITMO