Mexican Standoff (dessert)

Edoardo is organising the traditional New Year's Eve dinner of the *OIS* staff: after a round of well-stuffed pizzas, it will be desserts time! As everybody knows, this is a very delicate moment: everybody likes to order a dessert, but feels ashamed to do it. Thus, it is very easy to create a "Mexican standoff", where everyone is waiting for the others to take their move.

Figure 1: To dessert or not to dessert? That is the question.

More precisely, N staff members are attending the dinner. Member i (for $i = 0 \ldots N-1$) has M_i esteemed friends among the others,[1] which we call $F_{i,j}$ for $j = 0 \ldots M_i - 1$. Member i will feel allowed to take the dessert if and only if at least L_i of his esteemed friends have already done the same.

After the waiter asks for desserts, members that are willing to order one will do so, possibly making other members order one for themself, until everybody that has not ordered a dessert does not feel allowed to do so. Help Edoardo plan the dinner budget by determining how many staff members will order a dessert!

> ☞ Among the attachments of this task you may find a template file `dessert.*` with a sample incomplete implementation.

Input

The first line contains the only integer N. The description of the N members follows, each consisting of two lines: the first containing the two integers M_i, L_i, the second containing the M_i integers $F_{i,j}$.

Output

You need to write a single line with an integer: the number of staff members ordering a dessert.

[1] There are numerous rivalries in the OIS staff... their are not all friends with each other!

Constraints

- $2 \leq N \leq 300\,000$.
- $1 \leq M = \sum_{i=0}^{N-1} M_i \leq 300\,000$.
- $0 \leq L_i \leq M_i < N$ and $M_i \geq 1$ for each $i = 0 \ldots N-1$.
- $0 \leq F_{i,j} \leq N-1$, $F_{i,j} \neq i$ for each $i = 0 \ldots N-1$, $j = 0 \ldots M_i - 1$.
- There are no repeated friends: $F_{i,j} \neq F_{j,k}$ for $j \neq k$.
- Friendship is not symmetric: if i is a friend of j, j doesn't need to be a friend of i.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) ■□□□□	Examples.
– **Subtask 2** (20 points) ■■■□□	$M_i = 1$ for each $i = 0 \ldots N-1$.
– **Subtask 3** (10 points) ■■■□□	$L_i \leq 1$ for each $i = 0 \ldots N-1$.
– **Subtask 4** (40 points) ■■□□□	$N, M \leq 1000$.
– **Subtask 5** (30 points) ■■■□□	No additional limitations.

Examples

input	output

`2`
`1 1`
`1`
`1 0`
`0`
	`2`

`3`
`2 0`
`2 1`
`1 1`
`2`
`2 2`
`0 1`
	`1`

Explanation

In the **first sample case**, staff members 0 and 1 are friends with each other. Staff member 1 immediately orders a dessert, since he doesn't need any friends to have already ordered a dessert in order to feel allowed to do the same. After that, staff member 0 feels allowed to order a dessert as well.

In the **second sample case**, staff member 0 doesn't wait for anybody and immediately orders a dessert. Then, staff members 1 and 2 are both blocked waiting for each other to order a dessert: none of them will order one.