
optimal-agreement

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **256 megabytes**

We know how looking for pg's can be difficult. Both the tenant and the landlord must come to a mutual agreement. There are given n landlords and n tenants. Each tenant ranks all landlord in order of preference (first choice, second choice, and so on). Similarly, each landlord sorts all tenant according to his preference. The goal is to arrange n agreements in such a way that if a landlord 'l' prefers some tenant 't' more than his current tenant, and 't' prefers 'l' more than their current landlord, a new agreement occurs between l and t.

If t prefers their current landlord more, then they keep abiding the agreement, and no new agreement takes place. This problem always has a solution and your task is to find one.

Input

The first line contains a positive integer $t \leq 100$ indicating the number of test cases. Each test case is an instance of the problem defined. The first line of each test case is a positive integer $n \leq 500$ (the number of agreements to find). The next n lines are the tenant's preferences: ith line contains the number i (which means that this is the list given by the ith tenant) and the ordered list of preferred landlord (the first choice of ith tenant, the second choice,...). Then, the landlord's preferences follow in the same format.

Output

For each test case print n lines, where each line contains two numbers l and t, which means that the landlord number l and the tenant number t should have an agreement.

Example

standard input	standard output
1	1 3
4	2 2
1 4 3 1 2	3 1
2 2 1 3 4	4 4
3 1 3 4 2	
4 4 3 1 2	
1 3 2 4 1	
2 2 3 1 4	
3 3 1 2 4	
4 3 2 4 1	