Problem C. DFS Order 3

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Little Cyan Fish has a tree with n vertices. Each vertex is labeled from 1 to n. Now he wants to start a depth-first search at each vertex x. The DFS order is the order of nodes visited during the depth-first search. A vertex appears in the j-th ($1 \le j \le n$) position in this order means it is visited after $j-1$ other vertex. Because sons of a node can be iterated in arbitrary order, multiple possible depth-first orders exist.

The following pseudocode describes the way to generate a DFS order. The function GENERATE(x) returns a DFS order starting at vertex x:

Algorithm 2 An implementation of depth-first search

```
procedure DFS(vertex $x$)
    Append $x$ to the end of dfs_order
    for each son $y$ of $x$ do                ▷ Sons can be iterated in arbitrary order.
        DFS($y$)                              ▷ The order might be different in each iteration.
    end for
end procedure
procedure GENERATE($x$)
    Root the tree at vertex $x$
    Let dfs_order be a global variable
    dfs_order ← empty list
    DFS($x$)
    return dfs_order
end procedure
```

Let D_i be the returned array after calling GENERATE(x). Little Cyan Fish wrote down all the n sequences $D_1, D_2, \cdots, D_n$. Years later, he can no longer remember the structure of the original tree. Little Cyan Fish is wondering how to recover the original tree by using these n sequences. Please help him!

Input

There are multiple test cases. The first line contains one integer T ($1 \le T \le 10^5$), representing the number of test cases.

For each test case, the first line contains one positive integer n ($1 \le n \le 1\,000$), indicating the number of vertices of the tree.

The next n lines describe the DFS order of the original tree. In the i-th line of htese lines contains n integers $D_{i,1}, D_{i,2}, \cdots, D_{i,n}$, describes a DFS order. It is guaranteed that $D_{i,1} = i$ and D_i is a valid DFS order of the original tree.

It is guaranteed that the sum of n^2 over all test cases does not exceed 2×10^6.

Output

For each test case, you need to output $n-1$ lines, that describes the tree you recovered. In each of the $n-1$ lines, you need to output two integers u_i and v_i ($1 \le u_i, v_i \le n$), which means there's an edge between vertex u_i and vertex v_i. If there are multiple possible solutions, you may print any of them. It is guaranteed that at least one solution exists.

Example

standard input	standard output

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