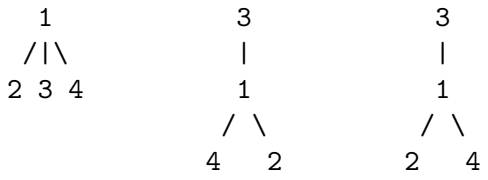
3. Turning a Tree (puu)

1 second 30 points

You are given a tree with nodes numbered $1 \ldots N$, where node 1 is the root of the tree and for each node the list of its child nodes is known.[1]

Find the tree we would get by lifting the leaf K of the original tree to be the new root, but leaving all edges intact, including the relative ordering of the edges at each node.

For example, starting from the tree shown on the left in the figure below and making the leaf 3 the new root, we would get the tree shown in the middle in the figure. The three shown on the right in the figure would not be correct answer, because the neighbors for the node 1 (listed counter-clockwise) are 2, 3, 4 in the original tree, but 2, 4, 3 in this tree.



Input. The first line of the text file `puusis.txt` contains the number of nodes N ($1 \le N \le 10{,}000$) and the index K of the leaf to become the new root ($1 \le K \le N$). The following N lines describe the structure of the original tree. The $(i+1)$-th line first contains m_i, the number of child nodes of the node i, and then the indices of the m_i child nodes, listed from left to right.

Output. The text file `puuval.txt` should contain exactly N lines: the structure of the new tree, in the format used in the input file.

Example.

puusis.txt	puuval.txt
4 3	2 4 2
3 2 3 4	0
0	1 1
0	0
0	

Explanation of the output lines:

1. Node 1 has 2 children, nodes 4 and 2 (in this order).
2. Node 2 has no children.
3. Node 3 has 1 child, node 1.
4. Node 4 has no children.

Grading. In test cases worth 16 points in total, the input is a binary tree (no node of the original tree has more than 2 children).

[1] See also `http://en.wikipedia.org/wiki/Tree_(data_structure)`