

Trees and Criticality

Time Limit: 2.0s **Memory Limit:** 256M

You are given an unrooted binary tree.

Let us define a concept called **criticality** for nodes. To calculate the criticality of a node, you must first remove that node along with any edges that have that node as an endpoint. The product of the numbers of nodes in each resulting closed component is the criticality of the node that was removed.

You are asked to find the highest criticality value among all the nodes in the tree.

Input

The first line contains one integer, n — the number of nodes in the tree.

Each of the next $n - 1$ lines contain two integers u and v , meaning that there is an edge between nodes u and v in the tree.

- $2 \leq n \leq 10^5$
- $1 \leq u, v \leq n$

Output

Print the highest criticality value among all the nodes in the tree.

Example

Input 1:

```
3
3 1
1 2
```

Output 1:

```
2
```

Input 2:

6
1 2
2 3
3 4
4 5
5 6

Output 2:

6

Explanation

Input 1:

A node of highest criticality is node 3, with a criticality value of 2.

Input 2:

A node of highest criticality is node 4, with a criticality value of $3 \cdot 2 = 6$.