D. Distance in Tree

A *tree* is a connected graph that doesn't contain any cycles.

The *distance* between two vertices of a tree is the length (in edges) of the shortest path between these vertices.

You are given a tree with n vertices and a positive number k. Find the number of distinct pairs of the vertices which have a distance of exactly k between them. Note that pairs (v, u) and (u, v) are considered to be the same pair.

Input

The first line contains two integers n and k ($1 \le n \le 50000$, $1 \le k \le 500$) — the number of vertices and the required distance between the vertices.

Next n - 1 lines describe the edges as " a_i b_i" (without the quotes) ($1 \le a_i, b_i \le n$, $a_i \ne b_i$), where a_i and b_i are the vertices connected by the i-th edge. All given edges are different.

Output

Print a single integer — the number of distinct pairs of the tree's vertices which have a distance of exactly k between them.

Please do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
5 2 1 2 2 3 3 4 2 5

output
4

input
5 3 1 2 2 3 3 4 4 5

output
2

Note

In the first sample the pairs of vertexes at distance 2 from each other are (1, 3), (1, 5), (3, 5) and (2, 4).