

E. Close Vertices

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You've got a weighted tree, consisting of n vertices. Each edge has a non-negative weight. The length of the path between any two vertices of the tree is the number of edges in the path. The weight of the path is the total weight of all edges it contains.

Two vertices are close if there exists a path of length at most l between them and a path of weight at most w between them. Count the number of pairs of vertices v, u ($v < u$), such that vertices v and u are close.

Input

The first line contains three integers n, l and w ($1 \leq n \leq 10^5, 1 \leq l \leq n, 0 \leq w \leq 10^9$). The next $n - 1$ lines contain the descriptions of the tree edges. The i -th line contains two integers p_i, w_i ($1 \leq p_i < (i + 1), 0 \leq w_i \leq 10^4$), that mean that the i -th edge connects vertex $(i + 1)$ and p_i and has weight w_i .

Consider the tree vertices indexed from 1 to n in some way.

Output

Print a single integer — the number of close pairs.

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
4 4 6 1 3 1 4 1 3
output
4

input
6 2 17 1 3 2 5 2 13 1 6 5 9
output
9