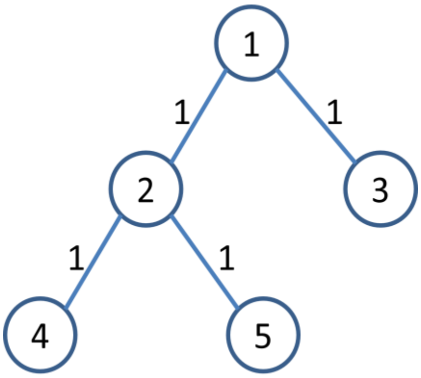
Diameter

Given an **N** node (nodes are numbered 1,2,…,**N**) edge-weighted tree and an integer **K**, you are asked to find a **K**-partitioning of the nodes so that maximum diameter is minimized. **K**-partitioning is the partitioning of the nodes into **K** partitions. *Diameter of partition* is defined as the maximum pairwise distance within the nodes of the partition. *Diameter of partitioning* is defined as the maximum of diameter of partition over all **K** partitions. Your task is to find a **K**-partitioning which minimizes the diameter of partitioning.

In any partitioning, each node in the tree belongs to exactly one partition and each partition has at least one node, i.e. partitions are non-overlapping, non-empty and collectively exhaust the nodes of the tree.

Example



Let the tree has **N**=5 nodes with edge weights all equal to 1, and **K**=2. Consider 2-partioning with partitions {1, 3} and {2, 4, 5}. The diameter of the first partition is 1 and the diameter of the second is 2 (the distance between 4 and 5). Diameter of this partitioning is 2. Consider another 2-partitioning with partitions {1} and {2, 3, 4, 5}. The respective diameters are 0 and 3 (the distance between 3 and 4). The diameter of this partitioning is 3. If we consider all of the 2-partitionings, calculate their diameter of partitioning and take the minimum over all we will get the value 2. So, 2 is your output to this example.

Task

Please write a program that finds the minimum diameter of partitioning value over all **K**-partitionings. The input/output format is explained below with a sample.

input file (`Standard Input`)	output file (`Standard Output`)

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

Input format: The first line has two space-separated integers (**N** and **K,** in this order). Each of the next **N**-1 lines contains three space-separated integers “`i j w`”; it states that there is an edge between nodes `i` and `j` with weight `w`. The limits are: **N**≤260 000, 1≤**K**≤**N**, and edge weights 1≤`w`≤1 000.

Output format: One line with a single integer. The output is the minimum diameter of partitioning value over all **K**-partitionings.

Subtasks

Subtask 1 (10 points)

K=1

Subtask 2 (15 points)

K=2

Subtask 3 (30 points)

N≤110 000

Subtask 4 (45 points)

Given limits in the *Input format* section.

Implementation details

You have to submit only one file, called `diameter.c`, `diameter.cpp` or `diameter.pas`. The file implements your full program.