Problem G. One Path

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a tree T consisting of N vertices. Each edge has a positive integer weight.

You can perform the following operation on the given tree.

- Delete an edge from the graph, then add a new edge between any two distinct vertices. The weight of the new edge must be the same as the weight of the deleted edge. The resulting graph need not be a tree.

We define the weight of a path as the sum of the weights of the edges on the path. The distance between two vertices u and v is defined as the weight of the *shortest path* from u to v — having the minimum weight. If there is no such path, we define the distance as 0.

The weight of a graph is the maximum of the weights between any two vertices.

Your task is to find the largest weight of the graph that can be obtained by performing the operation exactly i times, for $i = 0, 1, \ldots, K$.

Input

The first line contains two space-separated integers, N and K.

The i-th of the following $N - 1$ lines contains three space-separated integers, u_i, v_i, and w_i, representing an undirected edge that connects two different vertices u_i and v_i with a weight of w_i.

It is guaranteed that the edges form a tree.

- $2 \le N \le 2000$
- $0 \le K \le 2000$
- $1 \le u_i < v_i \le N$ $(1 \le i \le N - 1)$
- $1 \le w_i \le 10^9$ $(1 \le i \le N - 1)$

Output

Output $K+1$ space-separated integers. The i-th integer should be equal to the largest weight of the graph that can be obtained by performing the operation exactly $i - 1$ times.

Examples

standard input	*standard output*
5 1 1 3 2 4 5 4 3 4 3 2 3 7	14 16
7 2 1 2 4 2 3 6 2 4 2 4 5 5 2 6 1 4 7 3	13 20 21