D1. Networking (easy version)

Difficulty: Easy **Time**: 3 s **Memory**: 1024 MB

by canin3

Note: The only difference between the easy and hard versions is that in the easy version, $n \leq 10^4$ while in the hard version, $n \leq 10^6$.

I don't know much about networking, but let's say Purdue's PAL network consists of n magical gizmos numbered $1, 2, \ldots, n$. Since IT is on a budget, their network topology is a weighted tree with $n - 1$ edges (u_i, v_i, w_i) for $1 \leq i \leq n - 1$. The i-th edge is between gizmos u_i and v_i, and takes w_i milliseconds to traverse.

After HammerWars registration crashed their network, they decided to add m extra bidirectional "shortcut" edges (u_i, v_i, w_i) for $n \leq i \leq n + m - 1$.

You have devices connected to the first k gizmos $1, 2, \ldots, k$ for some $2 \leq k \leq n$. Using all (tree and shortcut) edges, how long does it take to travel between every pair of your devices i, j for $1 \leq i < j \leq k$?

Input

The first line contains 3 integers n, m, k $(3 \leq n \leq 10^4, 1 \leq m, k \leq 100)$.

The i-th of the next $n - 1 + m$ lines contains integers u_i, v_i, w_i $(1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9)$, denoting that the i-th edge connects vertices u_i and v_i.

It is guaranteed that the first $n - 1$ edges form a tree. It is also guaranteed that there are no self-edges $(u_i \neq v_i)$ and all edges are between different pairs of vertices (each possible pair of u_i, v_i appears at most once in the edge list).

Output

Print k lines.

On the i-th line, output k space-separated integers $a_1, \ldots, a_k$, where a_j is the time it takes to travel from gizmo i to gizmo j (or 0 when $i = j$).

Sample

Input

```
7 5 4
2 1 5
1 4 9
1 7 6
2 5 7
7 3 3
2 6 2
2 3 8
3 4 10
4 6 5
3 1 2
6 1 10
```

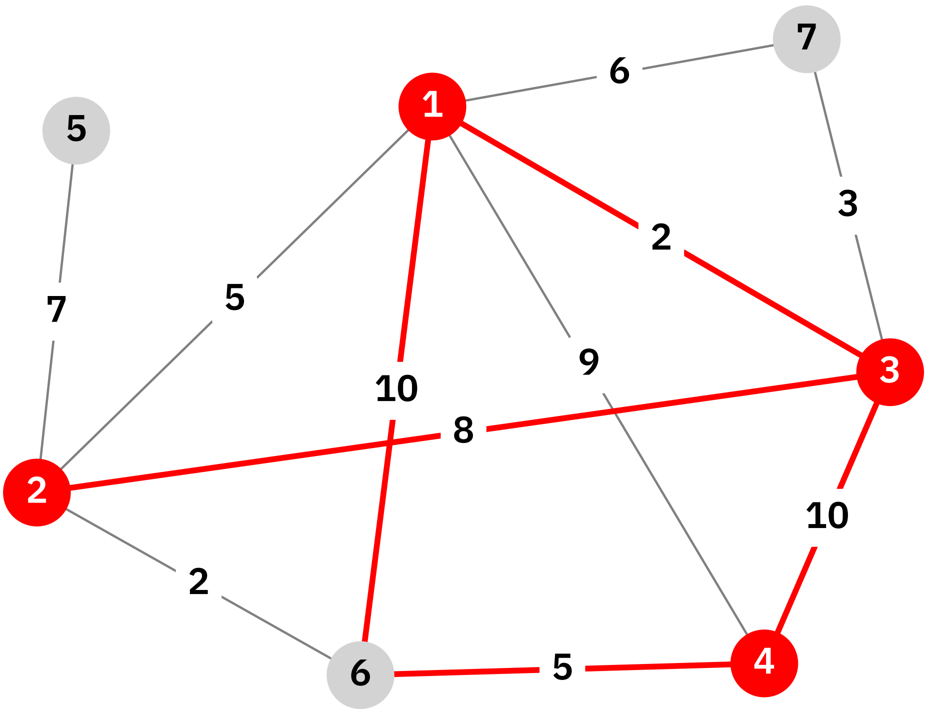
Output

```
0 5 2 9
5 0 7 7
2 7 0 10
9 7 10 0
```

Explanation

The next page contains a diagram of the graph for the sample. The first k gizmos and shortcut edges are highlighted in red.

For example, the 2nd element of the 3rd line is 7 since the shortest path between nodes 2 and 3 takes the shortcut edge with weight 2 from $3 \to 1$, then the tree edge with weight 5 from $1 \to 2$.


1
2
3
4
5
6
7
6
3
2
5
10
9
8
7
10
2
5