Горилла залезает на дерево

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Однажды горилла нашла дерево на n вершинах. Корнем этого дерева является вершина 1, а на рёбрах этого дерева записаны веса. Любая порядочная горилла захотела бы залезть на такое дерево и эта не стала исключением. Обратите внимание, что в этой задаче корень дерева находится наверху.

Горилла может начать свой путь в любой вершине с номером v ($2 \leqslant v \leqslant n$). Изначально её усталость равна 1. Когда горилла поднимается по ребру наверх (ближе к корню), её усталость **умножается** на вес этого ребра.

Горилла имеет выносливость k, поэтому она откажется идти по ребру, после которого её усталость станет **строго больше** k и **немедленно** закончит путешествие. Если такое ребро не встретилось, горилла закончит путешествие в корне.

Назовём *продолжительностью* путешествия количество рёбер, по которым прошла горилла. Обозначим *продолжительность* путешествия из вершины v как d_v.

Помогите горилле узнать числа $d_2, d_3, \ldots, d_n$. Другими словами, найдите *продолжительности* путешествия, если горилла будет начинать свой путь из вершины с номером v ($2 \leqslant v \leqslant n$).

Формат входных данных

В первой строке дано целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных.

Далее следует описание наборов.

В первой строке каждого набора даны целые числа n, k ($2 \leqslant n \leqslant 2 \cdot 10^5$, $1 \leqslant k \leqslant 10^9$) — количество вершин в дереве и выносливость гориллы.

В следующих $n-1$ строках каждого набора даны целые числа v, u, w ($1 \leqslant v, u \leqslant n$, $1 \leqslant w \leqslant k$) — концы соответствующего ребра в дереве и вес этого ребра.

Гарантируется, что заданный набор рёбер образует дерево. Также гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите $n-1$ целое число $d_2, d_3, \ldots, d_n$.

Пример

стандартный ввод	стандартный вывод
3 9 10 1 2 1 3 1 3 3 5 4 4 3 3 7 8 4 6 3 1 2 7 7 4 9 1 4 1 1 2 1 2 3 1 3 4 1 5 1000000000 2 1 1000000000 1 5 10 3 1 1 4 2 1000000000	1 1 2 1 2 2 1 3 1 2 3 1 1 1 1

Замечание

Дерево в первом наборе входных данных первого теста выглядит так:

