Problem C
Broken Minimum Spanning Tree
Time Limit: 1 Second, Memory Limit: 2G

Ethan was tasked with finding a minimum spanning tree of a weighted, connected, undirected graph. However, he misunderstood the task and found a spanning tree that may not be minimal. To make his spanning tree a minimum spanning tree, you perform a sequence of edge swaps. An edge swap removes one edge from the spanning tree and adds an edge from the graph which is not currently in the spanning tree. After each edge swap, the tree must still be a spanning tree. What is the minimum number of edge swaps you must perform to fix Ethan's minimum spanning tree?

Input

The first line of input contains two integers n ($2 \leq n \leq 2{,}000$) and m ($n-1 \leq m \leq 3{,}000$), where n is the number of nodes in the graph and m is the number of edges in the graph. The nodes are numbered from 1 to n.

Each of the next m lines contains three integers u, v ($1 \leq u, v \leq n, u \neq v$), and w ($1 \leq w \leq 10^9$), signifying an edge connecting nodes u and v with weight w. The edges are numbered from 1 to m.

It is guaranteed that the graph is connected. The first $n-1$ edges of the input are Ethan's initial spanning tree. The graph may not be simple; there can be multiple edges between the same pair of nodes.

Output

Output a single integer k, which is the minimum number of edge swaps needed to make the spanning tree a minimum spanning tree. Then output k lines, each with two integers a and b, where a is the number of the edge to remove and b is the number of the edge to add. If there are multiple sets of k edge swaps that work, any one will be accepted.

Sample Input 1	Sample Output 1
4 4 1 2 10 2 3 3 3 4 1 1 4 4	1 1 4

This page is intentionally left blank.