Everything under Control

When stardust refineries are not managed properly, resources are wasted and accidents may happen (e.g. the star oil spill in the year 3755 BBY that almost destroyed an entire galaxy). Since Oompa Loompas are the only species that are not affected by star radiation, all stardust refineries are currently run by them (back on Earth, they used to refine cocoa beans). Unfortunately, even though they are exceptional workers, Oompa Loompas are terrible at controlling.

In the Edge of Reason there is a stardust refinery that has been neglected for too long. It is descending into chaos and may soon cause a huge environmental disaster in that remote part of the universe. Your mission is to travel there (wearing your anti-radiation suit) and create a way for the local Oompa Loompa teams to control their projects methodically.

An Oompa Loompa project involves several tasks which must be carried out without interruptions. Some tasks may run in parallel, provided the given precedence relationships are fulfilled. The team must complete the project as soon as possible. Exact task durations are known. A project may be modeled by a network. There are several states, called *nodes*: one is the *initial node*, which represents the start of the project, another is the *final node*, which represents the end of the project, and the remaining nodes correspond to intermediate stages of the project execution. Each task is defined by a pair (i, j), where i and j are distinct nodes. Task (i, j) can start once all tasks with endpoint i have been finished. If it starts at instant t, it will be running in the interval $[t, t + d_{ij}[$ (right-side is open), where d_{ij} is its duration.

For example, in the network depicted in the figure below, there are 6 nodes and 9 tasks (whose durations are in red). The initial node is 1 and the final node is 6. Task $(3, 5)$ can start after tasks $(2, 3)$, $(1, 3)$ and $(4, 3)$ have all been finished. The least amount of time units needed to complete this project is 21: 12 units to complete all prerequisites for task $(3, 6)$, and then 9 time units to complete task $(3, 6)$.

$$\begin{tikzcd}
 & 2 \arrow[rr, "9"] \arrow[dr, "1"] & & 5 \arrow[dd, "2"] \\
1 \arrow[ur, "3"] \arrow[rr, "12"] \arrow[dr, "6"'] & & 3 \arrow[ur, "5"] \arrow[dr, "9"'] & \\
 & 4 \arrow[ur, "6"'] & & 6
\end{tikzcd}$$

Let us suppose that tasks can be postponed, provided that the total time spent on the project does not exceed the least possible amount (21 time units, for the project above). Which tasks could be postponed without deferring any subsequent task? Task $(2, 3)$ is one of them due to task $(1, 3)$ taking 12 time units, as is task $(2, 5)$. Tasks $(1, 3)$ and $(1, 2)$ cannot be postponed without delaying other tasks.

Task

Write a program that, given a network, finds the set of all tasks that can be postponed without deferring any subsequent task.

Input

The first line contains two integers, N and T, where N is the number of nodes and T is the number of tasks. Nodes are identified by integers, ranging from 1 to N. Nodes 1 and N are the initial and the final nodes, respectively. Each of the next T lines contains three positive integers: i, j and d_{ij}, where (i, j) defines a task and d_{ij} is its duration.

Constraints

$2 \leq N \leq 5\,000$	(Number of nodes)
$1 \leq T \leq 50\,000$	(Number of tasks)
$1 \leq d_{ij} \leq 100$	(Duration of a task)

Output

The output defines the set S of all tasks that can be postponed without deferring any subsequent task. The first line has an integer, k, which is the size of S. Each of the remaining k lines has two integers, i and j, representing that task (i, j) is in S. Tasks must appear in increasing order: $(i_1, j_1) < (i_2, j_2) \Leftrightarrow i_1 < i_2 \vee (i_1 = i_2 \wedge j_1 < j_2)$.

Sample Input

```
6 9
1 3 12
1 4 6
1 2 3
2 5 9
2 3 1
3 5 5
3 6 9
5 6 2
4 3 6
```

Sample Output

```
3
2 3
2 5
5 6
```