I'm a delivery boy! CPSPC 2006 Warszawa, Day 3

Submit:	`delivery.c` / `delivery.cpp` / `delivery.pas`	Time limit:	seconds
Input:	svio	Memory limit:	64 MB
Output:	svio	Points:	100

Hey, I'm a delivery boy, working for an intergalactic delivery company, delivering spare antennas to little green men and other vital things.

During the last month the price of rocket coal increased. And we want to save our galactic dollars. So we started searching for optimal journeys for our spaceships. Yeah, you are probably disappointed now, 'cause it's smells of a bloody easy problem. But only on the Earth. The problem with the planets is that they are still moving, so most of the distances are changing in time. However, as we already know from the Hubble's law, the universe is expanding linearly, so the distances also increase linearly.

Another problem is the galactic police. Yes, sometimes they disturb us with laws or something like that, so we should better avoid them. This means that we can use only some of the flight corridors between planets, which are known to be police-free.

We have a map of the Universe with N planets and M flight corridors. Each of the corridors is defined by a pair of planets it connects (all corridors are bidirectional) and a function telling the travelling distance through the corridor. The function is either a constant or a linear function of type $d + T$ where d is a constant and T is the current time.

And how do we calculate distance when travelling through multiple corridors? Well, it's easy – our ships travel faster than light (using the Theory of irrelativity), so they arrive at the exactly same moment as they have departed. So if we start our journey at time T, the distances depend only on the time T and they don't change during the flight.

Now, we need to find an optimal flight schedule. For a given pair of planets, we want to know what's the shortest path between them at every time. Therefore, you should find the path P_1 which is shortest at time $T_1 = 0$. As the distances change over time, this path ceases to be optimal at some time T_2, for which you should find a new shortest path P_2, which is optimal until time T_3 and so on.

Input

The input starts with four integers $x, y, N, T_{\max}$ where N $(1 \le N \le 250\,000)$ is the number of planets, x and y $(1 \le x, y \le N)$ are the planets between which you want to find the paths for times T such that $0 \le T \le T_{\max} \le 1\,000\,000$.

Then a description of all the corridors follows. Each corridor is described by four integers x_i, y_i, d_i, δ_i where x_i and y_i are the planets joined by the corridor $(1 \le x_i, y_i \le N)$, d_i $(1 \le d_i \le 1\,000\,000)$ is the distance at time $T = 0$ and δ_i is either 0 (the distance remains constant) or 1 (the distance is $d_i + T$ at time T). There are at most 1 000 000 corridors.

The input ends by a single -1.

Output

The output will be a sequence of lines describing the shortest paths. The i-th line will contain a number T_i (the time at which the previous path ceased to be shortest and it was replaced by this path; the time should be rounded to the nearest smaller integer) and a sequence of planets lying on the path.

In case no path exists between the planets x and y, you should print a single line with the sentence '`End of business.`' instead.

Restrictions

You can safely assume that $N \cdot (T_{\max} + \max_i d_i) \le 1\,000\,000\,000$ and that the output will contain at most 1 000 000 numbers.

Example

Input:

```
1 6 9 14
1 2 1 1
1 4 3 1
2 3 4 1
4 3 3 1
3 6 1 1
2 5 7 0
5 6 5 0
6 7 1 1
5 7 1 1
4 8 1 1
8 6 3 0
-1
```

Output:

```
0 1 2 3 6
2 1 4 8 6
7 1 2 5 6
```

Another example

Input:

```
3 2 4 100000
1 4 1 1
-1
```

Output:

```
End of business.
```