Simon the Spider

`spider.c`, `spider.cpp`, `Spider.java`

In the *Mosquito* problem, you have seen that the potential reproductive capabilities of insects are very high. Fortunately for us humans, the insects also have natural enemies that help to reduce their population. Spiders belong among the best known insect predators, so there is no wonder we included them in this problem set.

Simon the Spider ate many insects this summer and got fat. The threads of his web will soon be too thin to carry him and so he needs to reinforce them. Since fat spiders are also lazy, Simon wants to use as little material as possible — well, you know that spiders have to produce the material for building webs by themselves. Finally, he decided to reinforce only some of the threads, but it is important that every node of his web must be reachable over the reinforced links.

Additionally, Simon plans to spend his free time on one of the reinforced links and he wants this one link to be long. Therefore, when calculating the total length of all reinforced links, the length of the longest reinforced link will be *subtracted* instead of added. Help Simon to decide which links should be reinforced to have the lowest possible total length under these circumstances.

Input Specification

The input consists of descriptions of several webs. The first line of each description contains two numbers: the number N ($2 \leq N \leq 2000$) of nodes in the web and the number M ($0 \leq M \leq 1\,000\,000$) of the links between pairs of nodes. Each of the following M lines describes one link. The description of each link contains three positive integers u_i, v_i, ℓ_i, where u_i and v_i ($1 \leq u_i, v_i \leq N$ and $u_i \neq v_i$) are the two nodes connected by the link and ℓ_i is its length ($1 \leq \ell_i \leq 100\,000$).

Output Specification

For each web, print a single line with the minimum possible total length of reinforced links under all given conditions. Remember that the total length is the sum of the lengths of all reinforced links *minus* twice the length of the longest reinforced link.

If it is not possible to reach every node from every other node through a sequence of links, then print "`disconnected`" instead of the cost.

Sample Input

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

Output for Sample Input

```
disconnected
-1
```