Project 3: Battle Over Cities

It is vitally important to have all the cities connected by highways in a war. If a city is conquered by the enemy, all the highways from/toward that city will be closed. To keep the rest of the cities connected, we must repair some highways with the minimum cost. On the other hand, if losing a city will cost us too much to rebuild the connection, we must pay more attention to that city.

Given the map of cities which have all the destroyed and remaining highways marked, you are supposed to point out the city to which we must pay the most attention.

Input Specification:

Your program must read test cases from the standard input.

The input consists of several test cases. Each case starts with a line containing 2 numbers *N* (≤1000), and *M*, which are the total number of cities, and the number of highways, respectively. Then *M* lines follow, each describes a highway by 4 integers:

```
City1 City2 Cost Status
```

where City1 and City2 are the numbers of the cities the highway connects (the cities are numbered from 1 to N), Cost is the effort taken to repair that highway if necessary, and Status is either 0, meaning that highway is destroyed, or 1, meaning that highway is in use.

Note: It is guaranteed that the whole country was connected before the war.

The input ends with *N* being 0. That case must NOT be processed.

Output Specification:

For each test case, output to the standard output. Just print in a line the city we must protest the most, that is, it will take us the maximum effort to rebuild the connection if that city is conquered by the enemy.

In case there is more than one city to be printed, output them in increasing order of the city numbers, separated by one space, but no extra space at the end of the line. In case there is no need to repair any highway at all, simply output 0.

Sample Input:

```
4 5
1 2 1 1
1 3 1 1
2 3 1 0
2 4 1 1
3 4 2 0
4 5
1 2 1 1
1 3 1 1
2 3 1 0
2 4 1 1
3 4 1 0
4 5
1 2 1 1
1 3 1 1
2 3 1 0
2 4 1 1
3 4 2 1
0
```

Sample Output:

```
2
1 2
0
```

Grading Policy:

This assignment is due Wednesday, October 30th , 2013 at 10:00pm.

- **Programmer:** Write the program **(50 pts.) with sufficient comments**.
- **Tester:** Provide a set of test cases to fill in a test report **(20 pts.)**. Note that the tester is responsible, as well as the programmer is, for any bug later found by Judge. Write analysis and comments **(10 pts.)**.
- **Report Writer:** Write Chapter 1 **(6 pts.)**, Chapter 2 **(12 pts.)**, and finally a complete report **(2 pts. for overall style of documentation)**.