

Reliable Nets

Input: nets.in

Output: standard output

You're in charge of designing a campus network between buildings and are very worried about its reliability and its cost. So, you've decided to build some redundancy into your network while keeping it as inexpensive as possible. Specifically, you want to build the cheapest network so that if any one line is broken, all buildings can still communicate. We'll call this a *minimal reliable net*.

Input

There will be multiple test cases for this problem. Each test case will start with a pair of integers ($n \leq 15$) and ($m \leq 20$) on a line indicating the number of buildings (numbered 1 through n) and the number of potential inter-building connections, respectively. (Values of $n = m = 0$ indicate the end of the problem.) The following m lines are of the form $b_1 \ b_2 \ c$ (all positive integers) indicating that it costs c to connect building b_1 and b_2 . All connections are bidirectional.

Output

For each test case you should print one line giving the cost of a minimal reliable net. If there is a minimal reliable net, the output line should be of the form:

`The minimal cost for test case p is c .`

where p is the number of the test case (starting at 1) and c is the cost. If there is no reliable net possible, output a line of the form:

`There is no reliable net possible for test case p .`

Sample Input

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

Sample Output

`The minimal cost for test case 1 is 6.`

`There is no reliable net possible for test case 2.`