

11267 The 'Hire-a-Coder' Business Model

As some other websites, 'Hire-a-Coder' is a new website that works as an agent between custom software buyers and freelance software developers. According to the 'Hire-a-Coder' business model, a number of clients are registered in the system. Some of them are software buyers. They post custom software projects on the site. Then the coders bid on the projects and each of the projects is assigned to a single particular coder after the biddings and dealings. Any buyer may have more than one open project at a time. Similarly any coder can take up more than a single project at any time. But no coder is allowed to take multiple projects from the same buyer at a time. The site works as a medium in the total procedure. Now, if a coder successfully completes his assigned project and the buyer accepts his work, 'Hire-a-Coder' gets some commission for that. But if the coder fails to complete a project or satisfy the client with his work, the company has to pay a penalty to the buyer thus suffering a loss.

Recently it is found out that there are a number of fake coders that take up a project but never finish their work eventually causing the company a good amount of loss. So, they need someone to advise them which of the possible project assignments are to make and which are to avoid in order to minimize their loss.

Assume, all the registered clients of the company are denoted by a unique positive integer number between 1 and N . By the help of a great fortune-teller, they have already tabulated the future of assigning all possible buyer-coder pairs. You can assume that assigning any project from a particular buyer to a particular coder has the same future. In their tabulation, if a coder fails to do a project from a buyer, a positive integer value is assigned to this particular buyer-coder pair. The value indicates the amount of loss the company is going to suffer if they pair-up these two clients. On the other hand, in case of a successful completion of a project from a buyer-coder pair, a negative integer value is assigned to this pair which denotes negative loss i.e. amount of profit.

You are assured that, every single individual in the community has a communication link to all other people in this community, not necessarily through direct acquaintance. The link may well be via one or more other people. Also note that, if 'A' knows 'B' then 'B' always knows 'A'.

There's one more problem. You have the pairing information for all those N people but you don't know which of them are buyers and which of them are coders. But you know that, all the registered clients are either a buyer or a coder and there's no one who is a buyer and a coder simultaneously. If the data for any test case violates this assumption, print 'Invalid data, Idiot!' and stop processing that case further.

Now, bearing all these factors in mind, you are to select a subset of the possible pairings so that if the company makes strictly those assignments, their loss is minimum possible. Also, you have to choose the subset so that every single person still has a direct / indirect communication link to all other people.

Input

Every test case starts with a single positive integer N ($2 \leq N \leq 200$) which is the number of registered clients. The second line has another positive integer E ($1 \leq E \leq 11000$), the number of pairs tabulated by the fortune-teller. Each of the next E lines contains 3 integers A, B and C which means, if the company pairs A with B , they will end up in a loss with amount C . ($1 \leq A, B \leq N$) and $-1000 \leq C \leq 1000$). The last test case will be followed immediately by a single line containing a '0'. This line indicates the end of file.

Output

If there is any inconsistency in the data (as described in the statement) print ‘Invalid data, Idiot!’ in a single line. Otherwise, print the minimum possible loss the company can afford without disconnecting the social network.

Sample Input

```
6
9
1 4 1
2 4 2
3 4 3
1 5 2
2 5 3
3 5 2
1 6 3
2 6 1
3 6 2
6
10
1 2 3
1 4 1
2 4 2
3 4 3
1 5 2
2 5 3
3 5 2
1 6 3
2 6 1
3 6 2
4
4
1 3 5
1 4 -3
2 3 2
2 4 -6
0
```

Sample Output

```
8
Invalid data, Idiot!
-7
```