Toby is a great adventurer. Today he is trying to explore "Bitland" (a new country that will be remembered after Toby's exploration).

Bitland is divided into N small cities and M unidirectional roads between cities.

Toby begins the adventure at the city R, and after that he goes to any city R', if this new city (R') is not known by Toby, a road between R and R' is needed and he must pay a cost (in terms of adventure power) associated to the road. Otherwise, if Toby wants to go to a known city he does not need pay anything, even if there is no road from the current city to the target city (like teleportation)... is not Toby so cool?

Toby keeps traveling between cities until he reaches every city in Bitland. After this moment Toby goes to home, happy and eager for new adventures.

Wait! Where is the problem?

Did you remember that Toby has to pay for each road that is used to disclose a new city? Help Toby to minimize this cost (the sum of all power paid), because he needs as much energy as possible for his new adventures.

Input

The input starts with an integer $1 < T \leq 100$ indicating the number of test cases.

Each test case begins with three integers $3 < N \leq 10000$, $3 < M \leq N$, $0 \leq R < N$ denoting the number of cities, number of roads and initial city, respectively. Followed by M lines which contain three integers, $0 \leq u, v < N$, $1 \leq w \leq 10000$. These numbers denote a road from the city u to the city v with cost w.

Note that there could be several roads between the same pair of cities

Output

Print one line with the total cost for the adventure, followed by $N - 1$ lines with the chosen roads in the same format that was given in the input:

u v w — three space separated integers denoting a road from u to v with cost w.

If there are several answers, print any of them.

If there is no way to visit all the N cities, print 'impossible' without quotes.

Sample Input

```
2

5 5 0
0 1 1
0 2 100
1 3 2
3 2 3
2 4 4

5 5 4
0 1 1
0 2 100
1 3 2
3 2 3
2 4 4
```

Sample Output

```
10
0 1 1
3 2 3
1 3 2
2 4 4
impossible
```