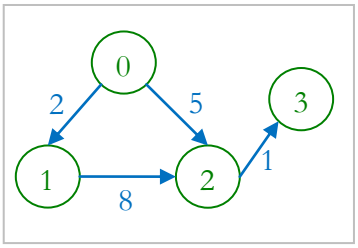
1380 – Teleport

You are in a tour and want to visit all the **n** cities in a country which are numbered from **0** to **n - 1**. Initially you are standing on city **K**. You have a car and can drive to one city to another. And you have a teleporting device, too. It can take you to any place to another place in no time. But the problem is that, the teleporting machine only works if the place you want to go is already visited by you. It can't detect the co-ordinates of a new city. But when you visit a city, the device locates the co-ordinates carefully, and thus helps you to teleport to this city from any other place. And the device is so strong that you can even teleport using the device along with your car! Since the traffic of the country is so high, the Govt. has planned to use only **m** unidirectional roads. Now you are given the map of the city and the required times for you to drive the roads. You have to find the minimum possible time needed for you to visit all the cities.

For example, for the city in the picture, you are in city 0. At first you go to city 1, and it will take 2 units of time, then you can go to city 2 from city 1, but it will take 8 units of time. You can teleport to city 0 and then go to city 2, it will cost you 5. So, you can visit city 1 and 2 in 7 units of time. And you can visit all the cities in time unit 8.



Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case starts with a black line. Next line contains three integers: **n m K (1 ≤ n ≤ 1000, 0 ≤ m ≤ 10000, 0 ≤ K < n)**. Each of the next **m** lines contains three integers **u v w (0 ≤ u, v < n, u ≠ v, 1 ≤ w ≤ 10000)** meaning that there is a road from city **u** to city **v** and takes **w** units of time to use that road. There can be at most one road from a city **u** to another city **v**.

Output

For each case, print the case number and the minimum time needed to visit all the cities. If it's impossible to do so, print **"impossible"**.

Sample Input	Output for Sample Input
2 4 4 0 0 1 2 1 2 8 0 2 5 2 3 1 2 1 1 0 1 10	Case 1: 8 Case 2: impossible

Note

Dataset is Huge, use faster I/O Methods.