

FAILING COMPONENTS

As a jury member of the Best Architectural Planning Contest, you are tasked with scoring the reliability of a system. All systems entered in the contest consist of a number of components which depend on each other. The reliability of such a system depends on the damage done by a failing component. Ideally a failing component should have no consequences, but since most components depend on each other, some other components will usually fail as well.

Most components are somewhat resilient to short failures of the components they depend on. For example, a database could be unavailable for a minute before the caches expire and new data must be retrieved from the database. In this case, the caches can survive for a minute after a database failure, before failing themselves. If a component depends on multiple other components which fail, it will fail as soon as it can no longer survive the failure of at least one of the components it depends on. Furthermore no component depends on itself directly, however indirect self-dependency through other components is possible.

You want to know how many components will fail when a certain component fails, and how much time passes before all components that will eventually fail, actually fail. This is difficult to calculate by hand, so you decided to write a program to help you. Given the description of the system, and the initial component that fails, the program should report how many components will fail in total, and how much time passes before all those components have actually failed.

Input

On the first line one positive number: the number of test cases, at most 100. After that per test case:

- one line with three space-separated integers n , d and c ($1 \leq n \leq 10\,000$ and $1 \leq d \leq 100\,000$ and $1 \leq c \leq n$): the total number of components in the system, the number of dependencies between components, and the initial component that fails, respectively.
- d lines with three space-separated integers a , b and s ($1 \leq a, b \leq n$ and $a \neq b$ and $0 \leq s \leq 1\,000$), indicating that component a depends on component b , and can survive for s seconds when component b fails. In each test case, all dependencies (a, b) are unique.

Output

Per test case:

- one line with two space-separated integers: the total number of components that will fail, and the number of seconds before all components that will fail, have actually failed.

Examples

Nº	stdin	stdout
1	2	2 5
	3 2 2	3 6

2 1 5	
3 2 5	
3 3 1	
2 1 2	
3 1 8	
3 2 4	