Homework Assignment #5: Programming Exercise #1

Note

This assignment constitutes 4% of your grade and is due 2:10PM Monday, May 10, 2010. Please write/type your answers/code on A4 (or similar size) paper. Drop your homework by the due time in Yih-Kuen Tsay's mail box on the first floor of Management College Building II. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers/code is strictly forbidden.

Your work will be graded according to its correctness and presentation. Specifically, you should demonstrate evidences showing that your program is correct. You should also organize and document your program in such a way that your classmates, for example, can understand it.

Problem

Solve either Problem C "Tracking Bio-bots" or Problem D "Castles" of the 2010 Annual ACM International Collegiate Programming Contest World Finals (see the appended).

Please prepare an input file with more interesting cases and test your program on the input. In the documentation of your program, you should describe how you have applied the algorithmic techniques, in particular design by induction, learned in class.

Problem C
Tracking Bio-bots
Problem ID: biobots

The researchers at International Bio-bot Makers (IBM) have invented a new kind of Bio-bot, a robot with behavior mimicking biological organisms. The development of the new robot is at a primitive stage; they now resemble simple four-wheeled rovers. And like most modern robots, Bio-bots are not very mobile. Their weak motors and limited turning capability put considerable limitations on their movement, even in simple, relatively obstacle-free environments.

Currently the Bio-bots operate in a room which can be described as an $m \times n$ grid. A Bio-bot occupies a full square on this grid. The exit is in the northeast corner, and the room slopes down towards it, which means the Bio-bots are only capable of moving north or east at any time. Some squares in the room are also occupied by walls, which completely block the robot. Figure 1, which corresponds to the sample input, shows an example of such a room.

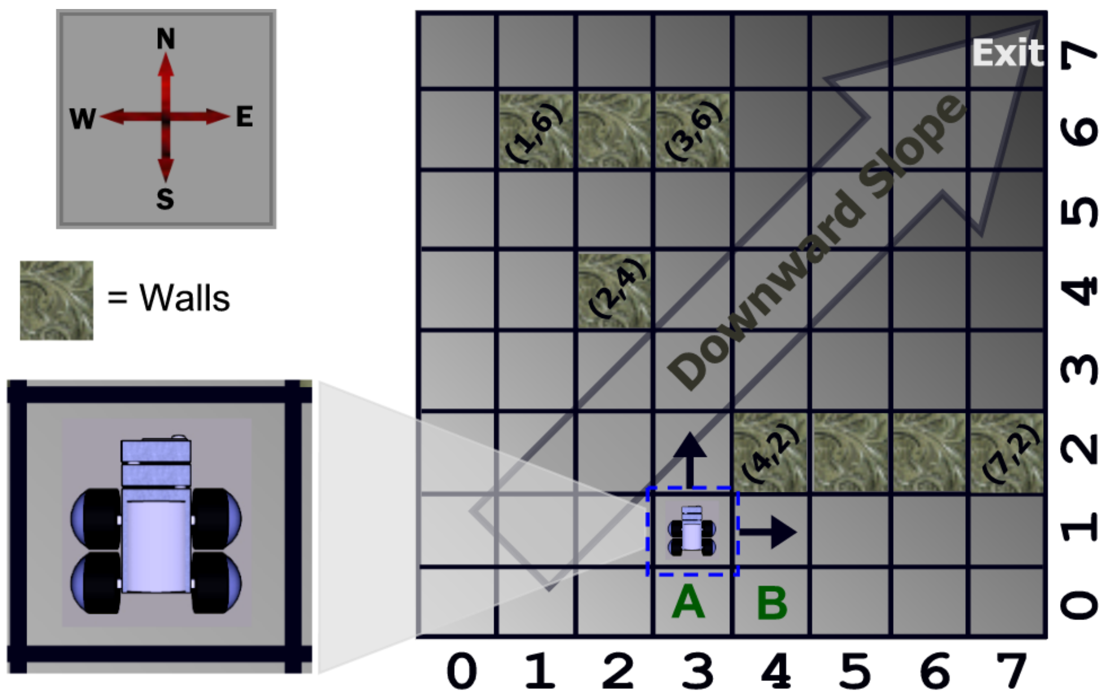

Figure 1

Clearly, a Bio-bot located on square A is capable of leaving the room, while one at square B is trapped inside it, no matter what it does. Locations like B are called "stuck squares." (Walls do not count as stuck squares.) Given the description of a room, your job is to count the total number of stuck squares in the room.

Input

Input consists of multiple test cases, each describing one room. Each test case begins with a line containing three integers m, n, and w ($1 \le m, n \le 10^6$, $0 \le w \le 1000$). These indicate that the room contains m rows, n columns, and w horizontal walls.

Each of the next w lines contains four integers x_1, y_1, x_2, y_2, the coordinates of the squares delimiting one wall. All walls are aligned from west to east, so $0 \le x_1 \le x_2 < n$ and $0 \le y_1 = y_2 < m$. Walls do not overlap each other. The southwest corner of the room has coordinates (0,0) and the northeast corner has coordinates (n -1, m -1).

The last test case is followed by a line containing three zeros.

Output

For each test case, display one line of output containing the test case number followed by the number of stuck squares in the given room. Follow the format shown in the sample output.

Sample Input	Output for the Sample Input
8 8 3 1 6 3 6 2 4 2 4 4 2 7 2 0 0 0	Case 1: 8

Problem D
Castles
Problem ID: castles

Wars have played a significant role in world history. Unlike modern wars, armies in the middle ages were principally concerned with capturing and holding castles, the private fortified residences of lords and nobles. The size of the attacking army was an important factor in an army's ability to capture and hold one of these architectural masterpieces.

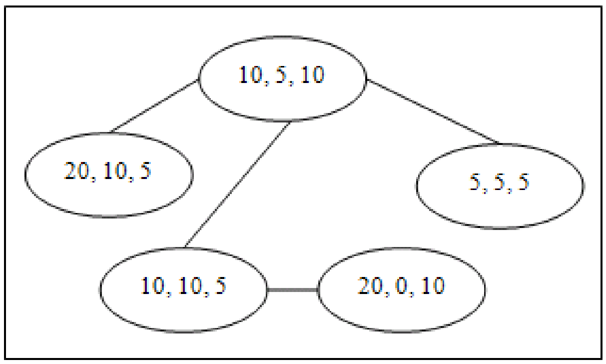

Figure 2

A certain minimum number of soldiers were required to capture a castle. Some soldiers were expected to die during the attack. After capturing the castle, some soldiers were required to remain in the castle to defend it against attacks from another enemy. Of course, those numbers were different for different castles. Commanders of the armies were obliged to consider the number of soldiers required for victory. For example, there are five castles in the region map shown in Figure 2. The castle at the lower right requires at least 20 soldiers to wage a winning attack. None are expected to perish during the attack, and 10 soldiers must be left in the castle when the army moves on.

In this problem you must determine the minimum size of an army needed to capture and hold all the castles in a particular region. For reasons of security, there is exactly one (bi-directional) route between any pair of castles in the region. Moving into the neighborhood of an uncaptured castle begins an attack on that castle. Any castle can serve as the first castle to be attacked, without regard for how the army got there. Once any castle has been captured, the requisite number of soldiers is left in the castle to defend it, and the remainder of the army moves on to do battle at another castle, if any remain uncaptured. The army may safely pass through the neighborhood of a castle that it has already captured. But because of the potential for attacks, the army may traverse the route between a pair of castles no more than twice (that is, at most once in each direction).

Input

The input contains multiple test cases corresponding to different regions. The description of the castles in each region occupies several lines. The first line contains an integer $n \le 100$ that is the number of castles in the region. Each of the next n lines contains three integers a, m, and g ($1 \le a \le 1000$, $0 \le m \le a$, $1 \le g \le 1000$), that give the minimum number of soldiers required to successfully attack and capture a particular castle, the number of soldiers that are expected to die during the attack, and the number of soldiers that must be left at the castle to defend it. The castles are numbered 1 to n, and the input lines describing them are given in increasing order of castle numbers. Each of the remaining $n - 1$ lines in a test case has two integers that specify the castle numbers of a pair of castles that are connected by a direct route.

A line containing 0 follows the description of the last region.

Output

For each test case, display the case number and the minimum number of soldiers in the army needed to conquer all the castles in the region. Follow the format shown in the sample output.

Sample Input	Output for the Sample Input
3 5 5 5 10 5 5 5 1 1 1 3 2 3 5 10 5 10 20 10 5 10 10 5 5 5 5 20 0 10 1 2 1 3 1 4 3 5 0	Case 1: 22 Case 2: 65