ICS 202 – Data Structures
Fall Semester 2018/2019 (181)
Assignment # 4
Due Saturday 8th December, 2018 before midnight

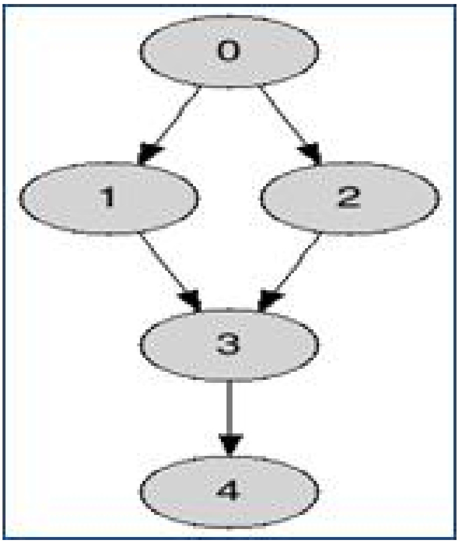
---

1. You need to submit the .java files for the programs.
2. Your programs should pass the test cases which are provided as separate files. Both the input and the expected output is provided so that you can check your programs for correctness.
3. You should select the most suitable data structures and taking into consideration the efficiency of the solutions using a particular data structure.
4. You are not supposed to use java libraries. You need to implement your own data structures to solve the problems.
5. There are 4 mandatory questions each carrying 25 points. The 5th and the 6th questions are bonus questions and they also carry 25 points, each.

Important Notes:

1. Your program should read from input files provided and write to the output files (with names provided to you) and NOT on the screen. Failure to do that will lead you to loose points.
2. You will be later (after few days) provided with hints to solve these problems but it is strongly encouraged to try solve them without using the hints.

QUESTION 1: INPUT FILE 'Q1.in', EXPECTED OUTPUT 'Q1.out'



In graph theory, a node X dominates a node Y if every path from the predefined start node to Y must go through X. If Y is not reachable from the start node then node Y does not have any dominator. By definition, every node reachable from the start node dominates itself. In this problem, you will be given a directed graph and you have to find the dominators of every node where the 0-th node is the start node.

As an example, for the graph shown right, 3 dominates 4 since all the paths from 0 to 4 must pass through 3. 1 doesn't dominate 3 since there is a path 0-2-3 that doesn't include 1.

Note: A 'node' is same as a 'vertex'.

Input

The first line of input will contain T (≤ 100) denoting the number of cases.

Each case starts with an integer N ($0 < N < 100$) that represents the number of nodes in the graph. The next N lines contain N integers each. If the j-th (0 based) integer of i-th (0 based) line is '`1`', it means that there is an edge from node i to node j and similarly a '`0`' means there is no edge.

Output

For each case, output the case number first. Then output $2N + 1$ lines that summarizes the dominator relationship between every pair of nodes. If node A dominates node B, output '`Y`' in cell (A, B), otherwise output '`N`'. Cell (A, B) means cell at A-th row and B-th column. Surround the output with '`|`', '`+`' and '`-`' to make it more legible. Look at the samples for exact format.

Sample Input

```
2
5
0 1 1 0 0
0 0 0 1 0
0 0 0 1 0
0 0 0 0 1
0 0 0 0 0
1
1
```

Sample Output

```
Case 1:
+---------+
|Y|Y|Y|Y|Y|
+---------+
|N|Y|N|N|N|
+---------+
|N|N|Y|N|N|
+---------+
|N|N|N|Y|Y|
+---------+
|N|N|N|N|Y|
+---------+
Case 2:
+-+
|Y|
+-+
```

QUESTION 2: INPUT FILE 'Q2.in', EXPECTED OUTPUT 'Q2.out'

John has n tasks to do. Unfortunately, the tasks are not independent and the execution of one task is only possible if other tasks have already been executed.

Input

The input will consist of several instances of the problem. Each instance begins with a line containing two integers, $1 \leq n \leq 100$ and m. n is the number of tasks (numbered from 1 to n) and m is the number of direct precedence relations between tasks. After this, there will be m lines with two integers i and j, representing the fact that task i must be executed before task j.

An instance with $n = m = 0$ will finish the input.

Output

For each instance, print a line with n integers representing the tasks in a possible order of execution.

Sample Input

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

Sample Output

```
1 4 2 5 3
```

QUESTION 3: INPUT FILE 'Q3.in', EXPECTED OUTPUT 'Q3.out'

In a certain city there are N intersections connected by one-way and two-way streets. It is a modern city, and several of the streets have tunnels or overpasses. Evidently it must be possible to travel between any two intersections. More precisely given two intersections V and W it must be possible to travel from V to W and from W to V.

Your task is to write a program that reads a description of the city street system and determines whether the requirement of connectedness is satisfied or not.

Input

The input contains several test cases. The first line of a test case contains two integers N and M, separated by a space, indicating the number of intersections ($2 \leq N \leq 2000$) and number of streets ($2 \leq M \leq N(N-1)/2$). The next M lines describe the city street system, with each line describing one street. A street description consists of three integers V, W and P, separated by a blank space, where V and W are distinct identifiers for intersections ($1 \leq V$, $W \leq N$, $V \neq W$) and P can be 1 or 2; if $P = 1$ the street is one-way, and traffic goes from V to W; if $P = 2$ then the street is two-way and links V and W. A pair of intersections is connected by at most one street.

The last test case is followed by a line that contains only two zero numbers separated by a blank space.

Output

For each test case your program should print a single line containing an integer G, where G is equal to one if the condition of connectedness is satisfied, and G is zero otherwise.

Sample Input

Note: 1 2 1 means a one-way street from '1' to '2'.

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

Sample Output

```
1
1
0
0
```

"A new internet watchdog is creating a stir in Springfield. Mr. X, if that is his real name, has come up with a sensational scoop."
`Kent Brockman`

There are n SMTP servers connected by network cables. Each of the m cables connects two computers and has a certain latency measured in milliseconds required to send an email message. What is the shortest time required to send a message from server S to server T along a sequence of cables? Assume that there is no delay incurred at any of the servers.

Input

The first line of input gives the number of cases, N. N test cases follow. Each one starts with a line containing n ($2 \leq n \leq 20000$), m ($0 \leq m \leq 50000$), S ($0 \leq S < n$) and T ($0 \leq T < n$). $S \neq T$. The next m lines will each contain 3 integers: 2 different servers (in the range $[0, n-1]$) that are connected by a bidirectional cable and the latency, w, along this cable ($0 \leq w \leq 10000$).

Output

For each test case, output the line '`Case #`x`:`' followed by the number of milliseconds required to send a message from S to T. Print '`unreachable`' if there is no route from S to T.

Sample Input

```
3
2 1 0 1
0 1 100
3 3 2 0
0 1 100
0 2 200
1 2 50
2 0 0 1
```

Sample Output

```
Case #1: 100
Case #2: 150
Case #3: unreachable
```

Saint Andrerw's Road, part of the Formula One circuit (Kenny Pek, Piccom)

Singapore will host a Formula One race in 2008. The race will be held on a 5.067km long street circuit, consisting of 14 left hand turns and 10 right hand turns. In the run up to the F1 race, the number of illegal night street racing activities have been on the rise. Such races consists of several rounds around a designated street circuit.

The authorities would like to deploy a new vehicle monitoring system in order to catch these illegal Saint Andrew's Road, part of the Formula One circuit racers. The system consists of a (Kenny Pek, Piccom) number of cameras mounted along various roads. For the system to be effective, there should be at least one camera along each of the possible circuits.

The Singapore road system can be represented as a series of junctions and connecting bidirectional roads (see Figure 5). A possible racing circuit consists of a start junction followed by a path consisting of three or more roads that eventually leads back to the start junction. Each road in a racing circuit can be traversed only in one direction, and only once.

Your task is to write a program that computes the optimal placement of the vehicle-monitoring cameras. You will be provided with a description of a connected road network to be monitored in terms of the roads and junctions. The junctions are identified by the bigger numbers in Figure 5. A camera can be deployed on the roads (and not the junctions).

The cost of deploying a camera depends on the road on which it is placed. The smaller numbers by the roads in Figure 5 indicate the cost of deploying a camera on that road. Your job is to select a set of roads that minimizes the total cost of deployment while ensuring that there is at least one camera along every possible racing circuit (i.e. loop in the road network).

Input

The input consists of a line containing the number c of datasets, followed by c datasets, followed by a line containing the number '0'.

The first line of each dataset contains two positive integers, n and m, separated by a blank, which represent the number of junctions and number of roads, respectively. You may assume that $0 < n < 10000$ and $0 < m < 100000$. For simplicity, we label each of the n junctions from 1 to n. The following m lines of each dataset each describes one road. Each line consists of three positive integers which are the labels of two different junctions and the cost of deploying a camera on this road. The cost of deploying a camera is between 1 and 1000.

Output

The output consists of one line for each dataset. The c-th line contains one single number, representing the minimal cost of setting up the vehicle monitoring system such that there is at least one camera along every possible circuit.

Note: The sample data set depicts the situation shown in Figure 5. The two cameras show where cameras might be placed in order to monitor each circuit at minimal cost. Since each of the cameras have a cost of 3, the total minimal cost is 6.

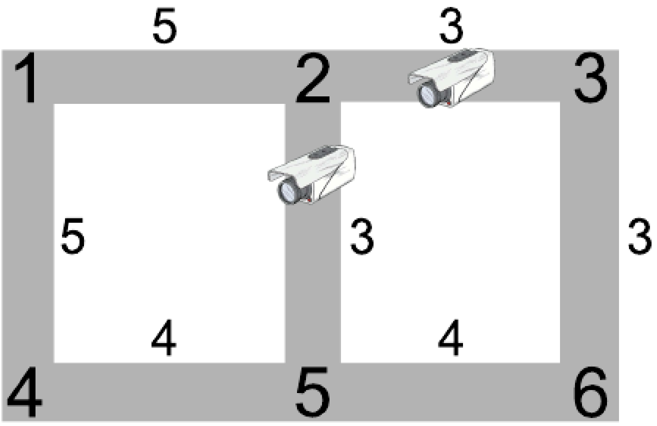


Figure 5: Illustration of sample input and one possible optimal placement of cameras

Sample Input

```
1
6 7
1 2 5
2 3 3
1 4 5
4 5 4
5 6 4
6 3 3
5 2 3
0
```

Sample Output

```
6
```

QUESTION 6: INPUT FILE 'Q6.in', EXPECTED OUTPUT 'Q6.out'

A construction company owns a large piece of real estate within the state of Florida. Recently, the company decided to develop this property. Upon inspection of the property, however, it was revealed that the land, at various locations, contained bodies of water. This came as a shock to the owners of the company, for they were from out of state and not familiar with wetlands of Florida. The situation was very grave and the owners not knowing that such bodies of water can be converted to beautiful lakes that will increase the value of the land around them, were about to abandon the construction project. Fortunately, this fact was brought to the owners' attention by a smart FIU graduate who worked for the company and consequently the construction project started.

The engineers divided the construction site by a grid into uniform square cells such that each square cell entirely contained either water or land. (How they did it, of course, is anybody's guess.) Now, the question that the engineers are to answer is the following: "Given the row and column number of a grid cell that contains water, what is the area of the lake containing that cell." (an area is measured by number of grid cells it contains. **Diagonal cells are considered to be adjacent**.)

You are to write a program to answer this question!

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The input consists of $0 < n \leq 99$ lines each containing $0 < m \leq 99$ character long sequence of 'L's and 'W's followed by $k > 0$ lines each containing a pair of integers i and j. The first n lines will represent the $n \times m$ grid covering the land where a 'W'/'L' at the c-th character of the r-th line indicates water/land within the cell at row r and column c of the grid. The pairs of integers on the last k lines, each represent the row and column numbers of some grid cell that contains water.

Note: row 'r' and column 'c' starts from 1 and not 0.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The output for each pair of integers, i and j, on the last k lines of input, consists of an integer, on a separate line, indicating the area of the lake containing the grid cell, at row i and column j of the grid.

Sample Input

```
1

LLLLLLLLL
LLWWLLWLL
LWWLLLLLL
LWWWLWWLL
LLLWWWLLL
LLLLLLLLL
LLLWWLLWL
LLWLWLLLL
LLLLLLLLL
3 2
7 5
```

Sample Output

```
12
4
```