Problem E

Connect the Campus

Input: standard input
Output: standard output
Time Limit: 2 seconds

Many new buildings are under construction on the campus of the University of Waterloo. The university has hired bricklayers, electricians, plumbers, and a computer programmer. A computer programmer? Yes, you have been hired to ensure that each building is connected to every other building (directly or indirectly) through the campus network of communication cables.

We will treat each building as a point specified by an x-coordinate and a y-coordinate. Each communication cable connects exactly two buildings, following a straight line between the buildings. Information travels along a cable in both directions. Cables can freely cross each other, but they are only connected together at their endpoints (at buildings).

You have been given a campus map which shows the locations of all buildings and existing communication cables. You must not alter the existing cables. Determine where to install new communication cables so that all buildings are connected. Of course, the university wants you to minimize the amount of new cable that you use.

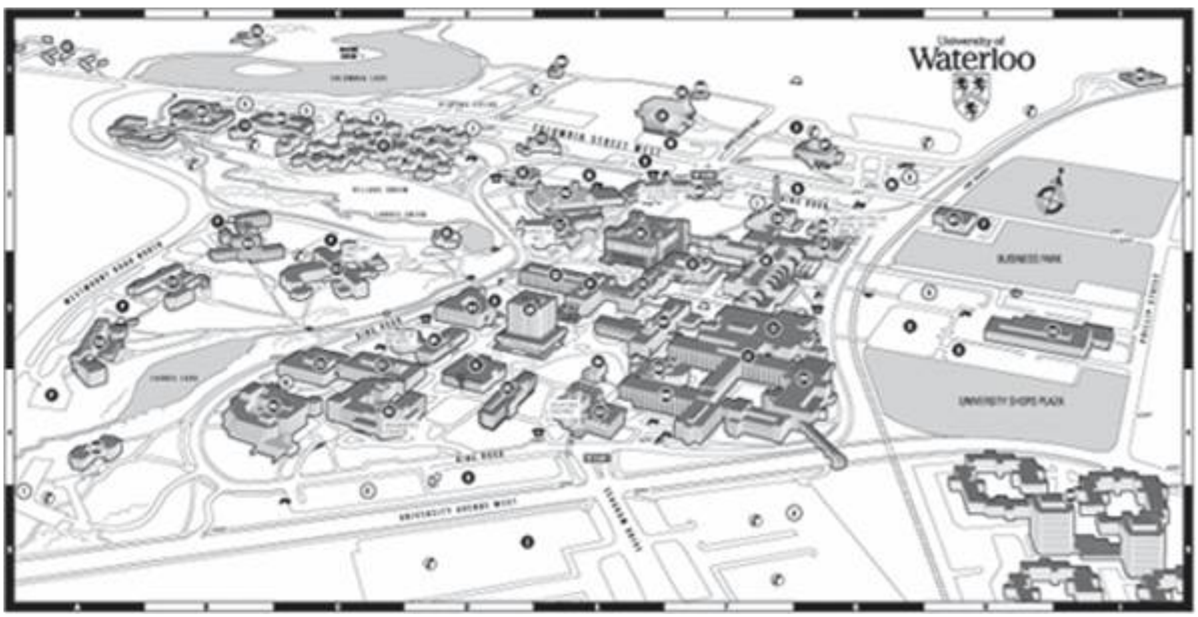


Fig: University of Waterloo Campus

Input

The input file describes several test case. The description of each test case is given below:

The first line of each test case contains the number of buildings **N (1<=N<=750)**. The buildings are labeled from **1** to **N**. The next **N** lines give the **x** and **y** coordinates of the buildings. These coordinates are integers with absolute values at most **10000**. No two buildings occupy the same point. After that there is a line containing the number of existing cables **M (0 <= M <= 1000)** followed by **M** lines describing the existing cables. Each cable is represented by two integers: the building numbers which are directly connected by the cable. There is at most one cable directly connecting each pair of buildings.

Output

For each set of input, output in a single line the total length of the new cables that you plan to use, rounded to two decimal places.

Sample Input

```
4
103 104
104 100
104 103
100 100
1
4 2
4
103 104
104 100
104 103
100 100
1
4 2
```

Sample Output

```
4.41
4.41
```

(Problem-setters: G. Kemkes & G. V. Cormack, CS Dept, University of Waterloo)

“A man running away from a tiger need not run faster than the tiger but run faster than the friend.”

Problem ?

Lift Hopping

Time Limit: 1 second

> Ted the bellhop: "*I'm coming up and if there isn't a dead body by the time I get there, I'll make one myself. You!*"
>
> `Robert Rodriguez, "Four Rooms."`

A skyscraper has no more than 100 floors, numbered from 0 to 99. It has **n** (1<=**n**<=5) elevators which travel up and down at (possibly) different speeds. For each **i** in {1, 2,... n}, elevator number **i** takes $\mathbf{T_i}$ (1<=$\mathbf{T_i}$<=100) seconds to travel between any two adjacent floors (going up or down). Elevators do not necessarily stop at every floor. What's worse, not every floor is necessarily accessible by an elevator.

You are on floor 0 and would like to get to floor **k** as quickly as possible. Assume that you do not need to wait to board the first elevator you step into and (for simplicity) the operation of switching an elevator on some floor always takes exactly a minute. Of course, both elevators have to stop at that floor. You are forbiden from using the staircase. No one else is in the elevator with you, so you don't have to stop if you don't want to. Calculate the minimum number of seconds required to get from floor 0 to floor **k** (passing floor **k** while inside an elevator that does not stop there does not count as "getting to floor **k**").

Input

The input will consist of a number of test cases. Each test case will begin with two numbers, **n** and **k**, on a line. The next line will contain the numbers $\mathbf{T_1}$, $\mathbf{T_2}$,... $\mathbf{T_n}$. Finally, the next **n** lines will contain sorted lists of integers - the first line will list the floors visited by elevator number 1, the next one will list the floors visited by elevator number 2, etc.

Output

For each test case, output one number on a line by itself - the minimum number of seconds required to get to floor **k** from floor 0. If it is impossible to do, print "IMPOSSIBLE" instead.

Sample Input	Sample Output
2 30 10 5 0 1 3 5 7 9 11 13 15 20 99 4 13 15 19 20 25 30 2 30 10 1 0 5 10 12 14 20 25 30 2 4 6 8 10 12 14 22 25 28 29 3 50 10 50 100	275 285 3920 IMPOSSIBLE

0 10 30 40 0 20 30 0 20 50 1 1 2 0 2 4 6 8 10	

Explanation of examples

In the first example, take elevator 1 to floor 13 (130 seconds), wait 60 seconds to switch to elevator 2 and ride it to floor 30 (85 seconds) for a total of 275 seconds.

In the second example, take elevator 1 to floor 10, switch to elevator 2 and ride it until floor 25. There, switch back to elevator 1 and get off at the 30'th floor. The total time is
10*10 + 60 + 15*1 + 60 + 5*10 = 285 seconds.

In example 3, take elevator 1 to floor 30, then elevator 2 to floor 20 and then elevator 3 to floor 50.

In the last example, the one elevator does not stop at floor 1.

Problemsetter: Igor Naverniouk
Alternate solutions: Stefan Pochmann, Frank Pok Man Chu

Problem I
FRIENDS

There is a town with N citizens. It is known that some pairs of people are friends. According to the famous saying that “The friends of my friends are my friends, too” it follows that if A and B are friends and B and C are friends then A and C are friends, too.

Your task is to count how many people there are in the largest group of friends.

Input

The first line of the input consists of N and M, where N is the number of town's citizens ($1 \le N \le 30000$) and M is the number of pairs of people ($0 \le M \le 500000$), which are known to be friends. Each of the following M lines consists of two integers A and B ($1 \le A \le N$, $1 \le B \le N$, $A \ne B$) which describe that A and B are friends. There could be repetitions among the given pairs.

Output

The output should contain one number denoting how many people there are in the largest group of friends.

Sample Input	Sample Output
2 2 2 1 2 2 3 10 12 1 2 3 1 3 4 5 4 3 5 4 6 5 2 2 1 7 10 1 2 9 10 8 9	3 6

Problem source: Bulgarian National Olympiad in Informatics 2003
Problem submitter: Ivaylo Riskov
Problem solution: Ivaylo Riskov

Problem D
Rings'n'Ropes
Time Limit: 3 seconds

"Well, that seems to be the situation. But,
I don't want that, and you don't want that,
and Ringo here definitely doesn't want that."
`Jules Winnfield`

I have **n** tiny rings made of steel. I also have **m** pieces of rope, all of exactly the same length. The two ends of each piece of rope are tied to two different rings.

I am going to take one of the rings, **L**, into my left hand, and another ring, **R** into my right hand. Then I will pull the whole structure apart as hard as I can. Some of the ropes will be streched horizontally because of this. Others will hang down or bend out of shape. If I want the number of horizontally stretched ropes to be as large as possible, which **L** and **R** should I pick?

Assume that the stretching of ropes in negligible, they all have negligible thickness and are free to slide around the rings that they are tied to. The thickness and radius of each ring is negligible, too.

Input
The first line of input gives the number of cases, **N**. **N** test cases follow. Each one starts with two lines containing **n** (2<=**n**<=120) and **m** (0<=**m**<=**n**(**n**-1)/2). The next **m** lines will each contain a pair of different rings (integers in the range [0, **n**-1]). Each pair of rings will be connected by at most one rope.

Output
For each test case, output the line containing "Case #**x**:", followed by the largest number of ropes that I can stretch horizontally by picking a pair of rings, **L** and **R**.

Sample Input	Sample Output
4 2 1 0 1 3 3 0 1 1 2 2 0 6 6 0 1 0 5 1 3 5 4 3 2 4 2	Case #1: 1 Case #2: 1 Case #3: 6 Case #4: 7

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

Problemsetter: Igor Naverniouk

558 Wormholes

In the year 2163, wormholes were discovered. A wormhole is a subspace tunnel through space and time connecting two star systems. Wormholes have a few peculiar properties:

- Wormholes are *one-way* only.
- The time it takes to travel through a wormhole is negligible.
- A wormhole has two end points, each situated in a star system.
- A star system may have more than one wormhole end point within its boundaries.
- For some unknown reason, starting from our solar system, it is always possible to end up in any star system by following a sequence of wormholes (maybe Earth is the centre of the universe).
- Between any pair of star systems, there is at most one wormhole in either direction.
- There are no wormholes with both end points in the same star system.

All wormholes have a constant time difference between their end points. For example, a specific wormhole may cause the person travelling through it to end up 15 years in the future. Another wormhole may cause the person to end up 42 years in the past.

A brilliant physicist, living on earth, wants to use wormholes to study the Big Bang. Since warp drive has not been invented yet, it is not possible for her to travel from one star system to another one directly. This *can* be done using wormholes, of course.

The scientist wants to reach a cycle of wormholes somewhere in the universe that causes her to end up in the past. By travelling along this cycle a lot of times, the scientist is able to go back as far in time as necessary to reach the beginning of the universe and see the Big Bang with her own eyes. Write a program to find out whether such a cycle exists.

Input

The input file starts with a line containing the number of cases c to be analysed. Each case starts with a line with two numbers n and m . These indicate the number of star systems ($1 \leq n \leq 1000$) and the number of wormholes ($0 \leq m \leq 2000$) . The star systems are numbered from 0 (our solar system) through $n-1$. For each wormhole a line containing three integer numbers x, y and t is given. These numbers indicate that this wormhole allows someone to travel from the star system numbered x to the star system numbered y, thereby ending up t ($-1000 \leq t \leq 1000$) years in the future.

Output

The output consists of c lines, one line for each case, containing the word `possible` if it is indeed possible to go back in time indefinitely, or `not possible` if this is not possible with the given set of star systems and wormholes.

Sample Input

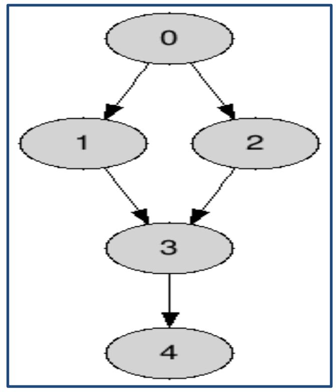
```
2
3 3
0 1 1000
1 2 15
2 1 -42
4 4
0 1 10
1 2 20
2 3 30
3 0 -60
```

Sample Output

```
possible
not possible
```

D	Dominator

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**.

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.

<table>
<tr><th>Sample Input</th><th>Output for Sample Input</th></tr>
<tr><td><pre>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</pre></td><td><pre>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|
+-+</pre></td></tr>
</table>

Problem Setter: Sohel Hafiz, Special Thanks: Kazi Rakibul Hossain, Jane Alam Jan