Algorithms for Programming Contests WS15/16 - Week 14

Chair for Efficient Algorithms (LEA), TU München

Prof. Dr. Harald Räcke

Moritz Fuchs, Philipp Hoffmann, Christian Müller, Chris Pinkau, Stefan Toman

This problem set is due by

Wednesday, 03.02.2016, 6:00 a.m.

Try to solve all the problems and submit them at

https://judge.in.tum.de/conpra/

This week's problems are:

All students are invited to join this contest. If you do not have an account yet register on the website given above or write a message to conpra@in.tum.de, we have additional accounts. You may work together in teams of two students. Please write a message to us if you want to have a team account.

Sample solutions, statistics and small prizes for the winners will be given on Wednesday, 03.02.2015, at 12:05 p.m. in room MI 00.08.038. Happy solving!

6 points will be awarded for each problem solved to contestants of the practical course.

If the judge does not accept your solution but you are sure you solved it correctly, use the “request clarification” option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge’s clarification form.

Problem A
Poker

As you know by now, Lea has always been interested both in games and statistics. Lately, she came to realize that there is no game as good at combining these two as Poker. She began reading up on strategy - soon things like “short stack”, “big stack”, “sharks” etc. began making sense to her. Her iron will to win awakened as she delved headfirst into the world of competitive poker.

She decided she would make a grand entrance to the scene by winning as much money as she could in a single month. This would be easy - she expected to be able to win all amateur tournaments easily with her newfound knowledge. Today, she had a look at the timetable for all the different amateur poker tournaments that would take place over the next months. Unfortunately, some of them overlapped, so she would not be able to win all of them. Can you tell her how much money she can win without going to overlapping tournaments?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing n, the amount of tournaments that take place in the next few months. n lines follow, one for each tournament. Each line consists of 3 numbers, a b p, where the tournament starts on day a, ends on day b and has a prize of p €for the winner. Two tournaments overlap even if the first one ends on the same day the second one starts.

Output

For each test case, print a line containing “Case #i: x” where i is its number, starting at 1 and x is the most amount of money Lea can win from all the tournaments (given that she wins each tournament she attends). Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 5000$
- $1 \leq a \leq b \leq 5000$
- $1 \leq p \leq 10000$

Sample Input 1	Sample Output 1
2 3 1 5 1000 10 15 1000 2 12 3000 4 1 1 500 2 5 500 3 7 1000 6 7 600	Case #1: 3000 Case #2: 1600

Sample Input 2	Sample Output 2
5 3 6 9 10 9 13 8 10 15 8 4 3 8 6 7 11 10 5 9 10 8 11 9 3 9 12 5 8 13 8 1 6 6 3 8 11 10 1 5 6 5 10 11 4 2 7 7 4 7 8 3 8 5 1 6 6	Case #1: 18 Case #2: 10 Case #3: 14 Case #4: 16 Case #5: 8

Problem B
Rumours

Rumours are a strong force. With their help one can achieve many things, good and bad. The minds of the masses can sometimes be steered in a profitable direction so that society may either bloom or collapse, depending on the message. Lea knows this well – and she intends to use it.

For now, Lea is interested in social experiments. She wants to see how quick and how far rumours will spread. So she starts to spread some rumour, waits a week, and then asks all people around her to rate their *information trust value* for this particular rumour, i.e. how strongly they believe the rumour to be a well-known fact.

Naturally, the information trust value has nothing to do with the actual truth of the rumour. Rather, it is based on who a person hears the rumour from – if it is someone they trust, they are more likely to believe it. Thus, the information trust value will always be the maximum *trustworthiness value* among the people they heard it from. Trustworthiness values range from 0 ("I would not even care if he/she died last week.") to 10 ("If he/she told me pigs fly, I would probably believe it.") and do not have to be symmetric. People will spread a rumour that they have heard if they believe it with some information trust value greater than 0.

So Lea went around and asked people for their trustworthiness values of all the other people around her. Then she started spreading the rumour that eating spaghetti after 1 a.m. would increase the risk of catching "Tomato Fever", where your head would take on the color of a tomato for a day or two.

A week later, she will ask all people for the information trust value for that particular rumour. Until then, she suspects that everybody will have had enough time to hear and spread the rumour. Alas, only a day has passed and Lea is already giddy for the results. Can you help her predict the total influence her rumour will have – i.e. the sum of all information trust values of all people?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing n, the number of people Lea knows the trustworthiness values of. n lines follow, with the i-th line containing n integers denoting the trustworthiness values $trust_{i,1}, \ldots, trust_{i,n}$ where $trust_{i,j}$ is the information trust value person i has in information heard from person j. Lea is person 1 and started spreading the rumour.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1 and x is the sum of all information trust values of all people. Lea started the rumour, so her information trust value for the rumour is 0. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 400$
- $0 \leq trust_{i,j} \leq 10$ for all $1 \leq i, j \leq n$
- $trust_{i,i} = 0$ for all $1 \leq i \leq n$

Sample Input 1

```
2
4
0 0 0 1
5 0 0 0
1 5 0 5
1 10 5 0

5
0 3 4 3 0
3 0 3 0 0
4 3 0 10 0
0 0 0 0 0
0 2 3 5 0
```

Sample Output 1

```
Case #1: 20
Case #2: 10
```

Sample Input 2

```
5
6
0 0 0 7 0 5
1 0 4 8 0 6
0 5 0 2 0 4
9 1 7 0 3 7
0 5 0 0 0 0
3 6 2 0 0 0

5
0 0 1 6 0
0 0 10 9 0
0 9 0 0 0
0 2 0 0 0
4 0 0 0 0

6
0 0 2 5 3 8
0 0 5 0 3 5
0 8 0 0 3 0
0 0 10 0 10 0
3 10 1 9 0 5
0 0 1 3 8 0

7
0 8 0 0 0 0 4
7 0 5 4 3 0 0
6 4 0 0 0 6 3
0 4 0 0 7 0 9
0 9 1 1 0 0 2
0 0 0 0 8 0 0
8 0 0 1 10 2 0

8
0 8 0 3 0 0 8 1
1 0 0 8 0 0 0 10
0 7 0 0 6 0 4 0
0 0 7 0 9 0 0 7
0 0 0 0 0 0 2 0
10 4 6 0 0 0 10 0
0 7 1 4 0 0 0 4
0 10 1 10 7 0 0 0
```

Sample Output 2

```
Case #1: 33
Case #2: 4
Case #3: 41
Case #4: 49
Case #5: 55
```

Problem C
The Programmer

Some years ago, it must have been around 2005, Lea met an eccentric guy who called himself *The Programmer*. He wore a red top hat and claimed to own a time machine as well as a magical screwdriver which was apparently able to solve all kinds of problems just by pointing at them. As it turned out his claim was valid and he really owned a time machine as well as a magic screwdriver. As he showed off his machine, he warned her not to touch any controls, *especially* not the big red button in the middle of the machine. Because *The Programmer* took a liking to her, he offered her to accompany him in his adventures in time and space. Naturally, she immediately accepted.

After several adventures in which they saved the universe multiple times, fought off evil creatures from other worlds and met a gigantic, yet delightful centipede, it was time for Lea to return to her home. However, while *The Programmer* was not looking, she could not resist to use her last chance for pushing the big red button. At the time – disappointingly – nothing happened and Lea went on with her normal life.

More than 10 years later, Lea wakes up as usual and finds the following words etched into her bedroom window:

HELP
Jumping uncontrollably
Unbound in time
Bound in space
WARN THEM

Lea immediately realizes that this must be connected to her pushing the big red button in *The Programmer*'s time machine over 10 years ago. She assumes, that the machine cannot be controlled anymore and hence *The Programmer* must be jumping through time randomly. However, apparently the machine only jumps around in a restricted area in space. As she searches the internet for sightings of the time machine, she finds a website called `http://www.thewatcherfiles.com/` which lists several coordinates as well as timestamps at which some object - which she recognizes to be the time machine - has been spotted. She assumes that all points between the sightings as well as points between endangered points have to be considered endangered. As the time machine could crash into some building at any point in time (it is a miracle this has not happened to this day!), she needs to warn people in the affected region that they are in danger!

Can you help Lea identify the endangered area?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing one integer n. n lines follow, each containing four integers x, y, z and d. x, y and z are coordinates in meters relative to some point $(0, 0, 0)$, d is the time coordinate in seconds since January 1, 1970, 00:00.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. The next line contains a number k, k lines follow. Each following line contains x and y - coordinates describing one vertex of the resulting polytope that describes the endangered area when viewed from above (i.e. like points on a 2-dimensional map). The lines should be ordered according to their x-coordinate. If the x-coordinates are equal, use the y-coordinates as tie-breaker.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$

- $0 \leq x, y, z \leq 10^6$
- $1111867200 \leq d \leq 1456790400$

Sample Input 1

```
2
4
0 0 5 1206819015
2 0 3 1370787905
0 2 2 1437771333
1 1 1 1206819015

6
0 0 1 1119068743
5 0 0 1396647891
2 2 6 1288114097
2 2 4 1312119233
0 5 2 1353372240
4 4 3 1132073132
```

Sample Output 1

```
Case #1:
3
0 0
0 2
2 0
Case #2:
4
0 0
0 5
4 4
5 0
```

Sample Input 2

```
3
5
4 3 6 1278834253
6 4 3 1200462317
6 5 3 1338036361
1 2 7 1451894583
7 1 4 1163900825

5
6 2 0 1401981377
3 7 1 1288451418
3 6 0 1117297905
3 3 5 1394979583
2 2 1 1216124543

6
1 1 6 1188646063
4 3 6 1278834253
6 4 3 1200462317
6 5 3 1338036361
1 2 7 1451894583
7 1 4 1163900825
```

Sample Output 2

```
Case #1:
3
1 2
6 5
7 1
Case #2:
3
2 2
3 7
6 2
Case #3:
4
1 1
1 2
6 5
7 1
```

Problem D
Private Key

Lea encrypts all her messages using PGP. A few minutes ago, Lea mistakenly sent her private key to a friend instead of the file she wanted to attach to the message. Lea needs to fix this mishap as soon as possible by generating a new key and revoking the old one. Lea started looking for the key revocation certificate she stored somewhere on her backup drive, but she has not found it yet. Then she remembers the folder where she placed the certificate, but since she spent quite some time scrolling through lists of file names she wants to get the file as fast as possible. Can you help her?

The backup drive consists of several folders and files, that are contained in each other in a tree structure. Lea may always go to a subdirectory of the current directory or go back to the previous directory of her file browser's history if there is one. If Lea wants to go to a subdirectory, follow a symbolic link, or open the certificate she has to read the list of contained files and directories of the current directory before executing the next step which takes some time. Going back to the last directory does not require her to read this list and therefore does not require any time. There are also symbolic links in the directories which can be thought of as directories having several parent directories.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a single line containing three space-separated integers n, x, and m, where n is the number of directories (labelled 1 to n), x is the directory the key revocation certificate is in, and m is the number of operations Lea executed already. Lea started in the root directory 1.

One line follows containing n space-separated integers $a_1, \ldots, a_n$, meaning that Lea needs a_i milliseconds to read all directories and files contained in directory i.

n lines follow describing the subdirectories and symbolic links. The i-th line starts with an integer b_i followed by b_i more integers, the numbers of the directories reachable as subdirectories or via symbolic links from directory i.

One line follows describing the operations Lea already executed. It contains m integers describing the operations in order. 0 means going to the previous directory, other numbers describe the index of the next directory chosen.

Output

For each test case, print a line containing "Case #i: y" where i is its number, starting at 1, and y is the minimum time Lea needs to find the certificate after asking for your help or "impossible" if there is no path to the certificate. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 3000$
- $1 \leq x \leq n$
- $1 \leq m \leq 10^4$
- $1 \leq a_i \leq 10^5$ for all $1 \leq i \leq n$
- $0 \leq b_i \leq n$ for all $1 \leq i \leq n$
- The given sequence of operations is always valid.

Sample Input 1

```
3
3 3 1
10 20 30
2 2 3
0
0
2

3 3 4
10 20 30
3 1 2 3
0
0
1 3 0 2

2 2 1
10 20
1 1
0
1
```

Sample Output 1

```
Case #1: 40
Case #2: 40
Case #3: impossible
```

Sample Input 2

```
3
7 1 10
95277 54119 75507 41267 63485 28122 62076
4 5 4 3 1
3 4 6 2
4 2 7 1 6
3 5 7 3
3 5 7 2
4 1 4 3 2
5 6 4 1 5 2
3 7 0 0 1 0 4 7 6 0

6 1 9
22895 78234 19687 31420 70628 53728
2 3 1
2 2 5
3 2 1 3
4 5 3 1 4
2 6 4
5 3 6 4 2 5
1 1 3 1 0 0 0 3 0

7 2 7
40921 90335 90071 83891 17761 15237 55162
2 5 7
2 5 1
3 2 4 5
3 4 6 2
2 7 1
4 4 7 3 6
3 7 1 4
5 1 0 7 1 0 4
```

Sample Output 2

```
Case #1: 95277
Case #2: 22895
Case #3: 174226
```

Problem E
Swordfighting

What a time to be alive. Lea just watched the new "Super Intense Sword Arena" movie. The whole movie was two glorious hours of intense sword dueling between a whole cast of different combatants that would do anything to crush their opponent. At the beginning of each fight, both combatants would be given a random sword from a huge assortment of long swords, broad swords, bastard swords, rapiers, estocs and the like (spoilers: the ones that were lucky enough to get a lightsaber won almost always).

At home, she looks for the frames she found the coolest – every time the two combatants had their swords locked and both tried to push the other one back. After Lea admired the stills for a while, she noticed something – some of the swords did not seem to be straight lines starting from the hilt but rather curved a little. Fascinated she began to investigate and measured several coordinates of the hilts of both swords in the frame. Also she measured the point where both swords clashed (where the sparks fly off).

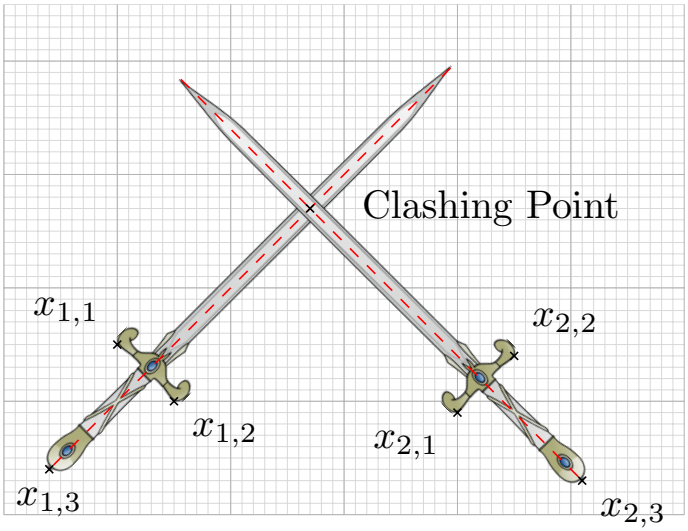

Figure E.1: Visual representation of the first case of the sample input. Sword Image taken from https://upload.wikimedia.org/wikipedia/commons/4/44/Long_sword.svg

Can you tell her if the clashing point is consistent – i.e. both swords seem to have straight blades?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing 12 integers $x_{1,1}$ $y_{1,1}$ $x_{1,2}$ $y_{1,2}$ $x_{1,3}$ $y_{1,3}$ $x_{2,1}$ $y_{2,1}$ $x_{2,2}$ $y_{2,2}$ $x_{2,3}$ $y_{2,3}$, where the first index specifies which sword it is and the second which part of the sword. Sword parts 1 and 2 are outer points of the crossguard at the same height (i.e. form a line orthogonal to the blade), while 3 is a point at the end of the pommel (see picture). Due to perspective and sword arena physics, you can ignore the width of the blades and think of the blade of the i–th sword as a single straight line extending from the crossguard. $(x_{i,3}, y_{i,3})$ always lies on an imaginary extension of the blade.

Output

For each test case, print a line containing "Case #i: x_c y_c" where i is its number, starting at 1 and (x_c, y_c) is the point where the two blades should clash with an error of up to 10^{-4}. If the blades of both swords do not clash, print a line containing "Case #i: strange". The hilt of a sword never clashes, only the blade. Even if two hilts overlap, it is still "strange". Both swords' blades are long enough to eventually clash. Each line of the output should end with a line

break.

Constraints

- $1 \leq t \leq 20$
- $0 \leq x_{i,j}, y_{i,j} \leq 100$ for all $i \in \{1, 2\}, j \in \{1, 2, 3\}$.
- $y_{i,3} < \min(y_{i,1}, y_{i,1})$ for all $i \in \{1, 2\}$.
- The projection of $(x_{i,3}, y_{i,3})$ onto $(x_{i,1}, y_{i,1}) - (x_{i,1}, y_{i,1})$ lies between $(x_{i,1}, y_{i,1})$ and $(x_{i,1}, y_{i,1})$ for $i \in \{1, 2\}$.
- $(x_{1,3}, y_{1,3}), (x_c, y_c)$ and $(x_{2,3}, y_{2,3})$ will never be collinear.

Sample Input 1

```
2
10 15 15 10 5 4 40 9 45 14 51 3
10 25 20 25 15 10 40 20 50 20 45 15
```

Sample Output 1

```
Case #1: 27.5 26.5
Case #2: strange
```

Sample Input 2

```
8
5 1 1 1 3 0 8 2 10 2 9 0
2 2 4 1 2 0 10 2 7 2 9 1
3 2 5 1 2 0 10 2 7 2 8 1
9 2 6 1 8 0 2 2 4 1 3 0
2 1 4 1 2 0 9 1 7 1 8 0
2 1 4 1 4 0 10 1 8 1 9 0
2 2 4 1 2 0 7 2 10 2 9 0
6 2 9 1 6 0 2 2 5 2 5 1
```

Sample Output 2

```
Case #1: strange
Case #2: 9.0 14.0
Case #3: 8.0 12.0
Case #4: 6.0 6.0
Case #5: strange
Case #6: strange
Case #7: 9.0 14.0
Case #8: strange
```

Problem F
Fast Food

Lea wants to meet her friend Bea at a new restaurant near the campus of the Thomas Underwood University (TUM). Since most of the customers are math students, the owner decided to print the place cards as follows: Each card contains the row a and column b of the table (tables are ordered in a perfect grid) as well as the greatest common divisor, largest common multiple and of course the prime factors of a and b.

Since simply telling your friend at which table you sit would be boring, some customers have invented the following fun game: Instead of telling your friend the exact table-coordinates, you send the gcd of the table you are sitting at as well as the gcds of a few tables to the right. I.e. if you are sitting at table (a, b), you will send $gcd(a, b), gcd(a + 1, b), \ldots, gcd(a + k, b)$ for some k. Your friend then has to figure out at which table you might sit. Because there might be multiple solutions, it has become a custom to tell the friend the right coordinates once he has come up with one correct solution.

Because most of the students communication is performed using Twottr, a short messaging service which allows only 39 letters per message, a and b can each have at most 19 digits.

Of course Bea takes part in this game and has sent Lea some gcds. Can you help Lea figure out where Bea might sit?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer k, the number of gcds. The next line contains k numbers $a_0, \ldots, a_{k-1}$, where $gcd(a, b) = a_0$, $gcd(a + 1, b) = a_1, \ldots, gcd(a + k - 1, b) = a_{k-1}$.

Output

For each test case, output one line containing "Case #i: a b" where i is its number, starting at 1, and a, b are integers s.t. $gcd(a, b) = a_0$, $gcd(a + 1, b) = a_1, \ldots, gcd(a + k - 1, b) = a_{k-1}$ and $1 \leq a, b \leq 10^{19}$.

Constraints

- $1 \leq t \leq 20$
- $3 \leq k \leq 1000$
- $1 \leq a_i \leq 10^{19} - k$

Sample Input 1

```
2
4
3 2 1 6

4
1 2 3 4
```

Sample Output 1

```
Case #1: 3 6
Case #2: 1 12
```

Sample Input 2

```
9
3
5 4 3

5
4 5 6 7 8

10
10 3 2 1 6 5 2 3 2 11

7
3 2 1 30 17 2 3

9
7 2 3 2 5 6 1 14 9

10
3 14 11 12 1 2 3 4 7 6

6
84 1 2 3 4 11

6
2 3 14 5 6 1

4
18 7 10 3
```

Sample Output 2

```
Case #1: 55 60
Case #2: 4 840
Case #3: 200 330
Case #4: 387 510
Case #5: 91 630
Case #6: 405 924
Case #7: 336 924
Case #8: 152 210
Case #9: 468 630
```

Problem G
The Mulk

This afternoon Lea went to an art showing by Cruce Canner in downtown New Tempolis. She did not really like the art and told the artist that she thinks it is pretentious and compared to M.N. Cut's work almost trivial (M.N. Cut is a very famous artist in Templonia). Canner did not take the critique very well, got angry and stormed out the door. A few minutes later, the news showed *The Mulk*, a big yellow raging monster, in front of the art gallery, at which point Lea remembered why the name Cruce Canner somehow sounded familiar.

Realizing that she was the only person in town who knew where *The Mulk* was headed (M.N. Cut's art gallery obviously), she called the town major and offered her assistance in catching *The Mulk*. He told her that they have a lot of roadblocks which can be deployed within seconds at any point in town (this is not their first Mulk-related incident), however, two rules need to be respected:

- Since the roadblocks are very expensive, as few of them as possible should be used - even at the expense of a few cars or houses. In particular this means, that we can let *The Mulk* run around for a bit before trapping him.
- Article 126.7a of the Mulk Defense Act of 2004 states that "in order not to disturb local wildlife too much, all roadblocks must be deployed at the same time".

Can you help Lea figure out how many roadblocks have to be used in the worst case to keep *The Mulk* from reaching M.N. Cut's gallery?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with two integers n and m. n is the number of road-crossings numbered from 1 to n on the map, m is the total number of roads between road crossings. m lines follow describing the map. Each line contains two integers u and v and describes a road between the crossings u and v.

The Mulk starts at crossing 1 and wants to reach crossing n.

Output

For each test case output one line containing "Case #i: x" where i is its number, starting at 1, and x is the number of roadblocks that need to be used to trap *The Mulk* in the worst case.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 120$
- $n \leq m \leq 10000$

Sample explanation

In the first sample input we are given 8 crossings with 9 roads in between them. The resulting graph looks this:

Now, we could decide to place roadblocks right away, for example as shown in figure G.2. This uses up 3 roadblocks.

However, if we allow *The Mulk* to run around a bit, we can see, that a better solution is possible: In the first step, he has to decide between going to node 2,3 or 4. When we place the roadblocks after *The Mulk* has made his first step, we can place them for example as shown in figure G.3, which results in only 2 roadblocks being used. Hence the correct answer is 2.

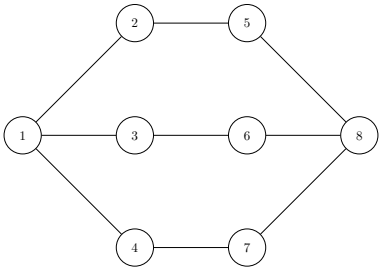

Figure G.1: Graph for the first sample input

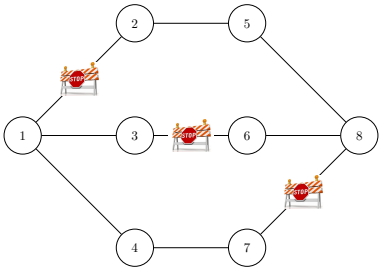

Figure G.2: Non-optimal way of placing the roadblocks.

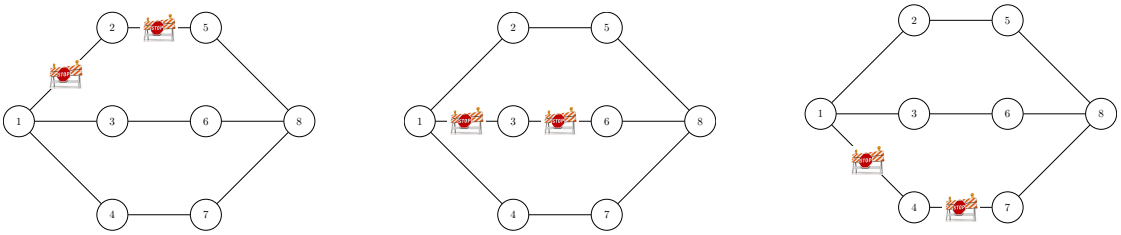

Figure G.3: Optimal roadblocks for all paths from node 1 to node 8.

Sample Input 1	Sample Output 1
3 8 9 1 2 1 3 1 4 2 5 3 6 4 7 5 8 6 8 7 8 5 7 1 5 1 2 1 4 5 3 5 4 5 2 3 4 2 2 1 2 1 2	Case #1: 2 Case #2: 3 Case #3: 2

This page is intentionally left blank.

Problem H
Sure Bet

Lea loves when others are betting on sports competitions of all sorts. She only enjoys looking at the odds, bets, payouts, scores, and statistical data in general. But a few friends of hers have now pushed her to actually make use of her knowledge and try to win some money. Lea is sure she can do that, but still has to ponder which bets to place. There are just too many competitions that she could bet on. The small bets cannot win you that much, this is why Lea wants to bet at least a certain amount of money. On the other side, she does not dare to bet too much. Can you help her find the correct competition for her to bet on?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two integers n m, the number of sports competitions n, and the minimum amount of money m Lea wants to spend. n lines follow, describing the different sports competitions. The i-th line consists of a string c_i and an integer b_i, meaning that the competition with name c_i has the minimum bet b_i.

Output

For each test case, output a line containing "Case #i: s" where i is its number, starting at 1, and s is either the name of a competition with minimal bet (but at least bet m), or "impossible" if there is no such competition. If there are multiple possible solutions, any of them will be accepted.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10000$
- $0 \leq m \leq 10000$
- $1 \leq b_i \leq 10000$ for all $1 \leq i \leq n$
- The c_i's only consist of lower case letters "`a-z`" and numbers "`0-9`".
- $c_i \neq c_j$ for all $1 \leq i, j \leq n$ and $i \neq j$
- $1 \leq |c_i| \leq 30$ for all $1 \leq i \leq n$

Sample Input 1

```
9
4 10
horserace 8
tennismatch 9
footballmatch 11
bikerace 9

5 10
basketballmatch 7
tennismatch1 25
footballmatch1 25
footballmatch2 6
tennismatch2 22

4 10
bikerace1 8
bikerace2 6
bikerace3 4
cricketmatch 5

6 100
formula1race 10
horserace1 110
horserace2 111
tennismatch1 110
footballmatch 111
tennismatch2 11

3 27
volleyballmatch1 204
marathon1 388
skirace1 538

3 52
volleyballmatch1 775
cricketmatch1 134
chessgame1 453

7 66
marathon1 934
cricketmatch1 353
marathon2 875
tennismatch1 758
basketballmatch1 9
formula1race1 784
basketballmatch2 653

3 19
footballmatch1 888
bikerace1 421
chessgame1 990

2 99
baseballmatch1 70
marathon1 10
```

Sample Output 1

```
Case #1: footballmatch
Case #2: tennismatch2
Case #3: impossible
Case #4: horserace1
Case #5: volleyballmatch1
Case #6: cricketmatch1
Case #7: cricketmatch1
Case #8: bikerace1
Case #9: impossible
```