Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Ernst W. Mayr
Christian Müller, Philipp Hoffmann, Chris Pinkau, Stefan Toman

Summer Term 2015
Problem Set 11
09.07.2015

Algorithms for Programming Contests

This problem set is due by

Thursday, 16.07.2015, 6:00 a.m.

Try to solve all the problems and submit them at

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

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 Thursday, 16.07.2015, at 12:30 p.m. in room MI 00.13.009A. Happy solving!

There will be 6 points awarded for each problem solved for 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.

SS15N11A Chocolate Tasting

Author: Stefan Toman

Lea is a well-known chocolate gourmet: She loves all types of chocolate and has an impressive knowledge about them. Next week, she wants to organize a big chocolate tasting for all of her friends. They should try even the most exotic chocolate bars and feel the incomparable pleasure of melting high-quality chocolate on their tongues.

As Lea has many friends and maybe even more kinds of chocolate she likes, she needs to plan buying the chocolate in advance. She will need quite a few trips to the store to buy all the chocolate she needs. Since it is a little warmer at her home than in the chocolate store she knows in which order to buy the chocolate in a way that each chocolate bar has the perfect eating temperature when the big chocolate tasting starts. To find the perfect order, one needs to have a great knowledge about melting temperatures and the temperature curves for making chocolate, but Lea knows everything by heart and computed that order already. The store also prepared packages for Lea containing the chocolate bars.

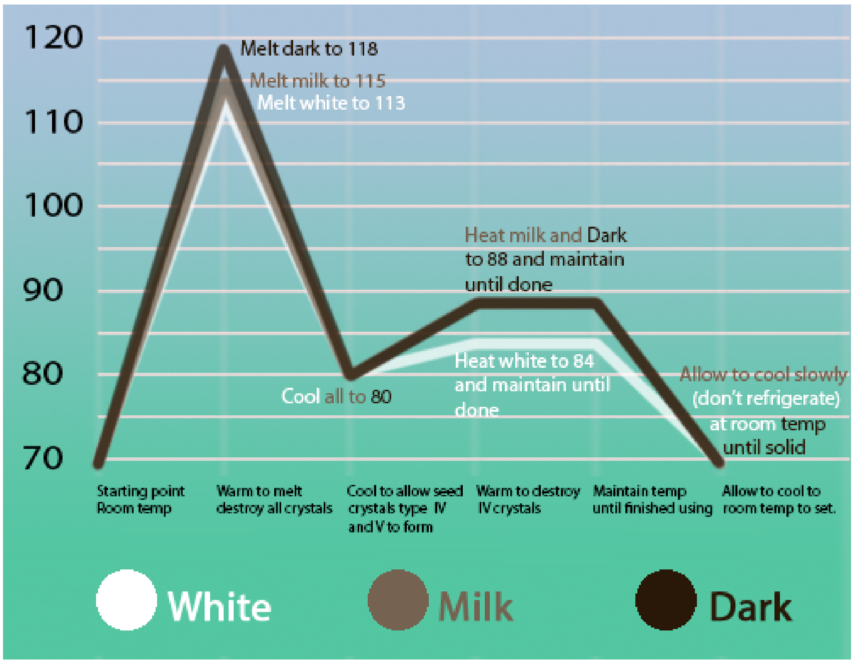

Figure 1: The perfect temperature curve for making chocolate.
Source: `http://www.thecookinggeek.com`

Given the order to buy the chocolate bars, she needs to decide which packages to take at each visit of the store. Lea wants to visit the store as few times as possible. If she reaches the minimum number of store visits, she wants to minimize her chocolate carrying coefficient CCC. The CCC of one trip to the store is computed as

$$CCC(x) = \begin{cases} 0 & \text{if } x \leq a \\ (x-a)^2 & \text{if } a < x \leq b \end{cases}$$

where x is the amount of chocolate she carries at the moment, a is the amount of chocolate that Lea can carry without problems and b is the maximum amount of chocolate Lea can carry at all (all given in kilograms). The sum of the CCC's of her trips should be as small as possible. Can you tell her what this sum will be?

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 three space-separated integers n, a and b where n is the number of chocolate packages and a and b are as described above. One line containing n space-separated integers follows describing the chocolate packages to buy in order. The i-th integer x_i describes the weight of the i-th package.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is the minimum sum of all CCC's such that the number of store visits is also minimized. Print "impossible" for y if Lea is not able to carry the chocolate to her home.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10^5$
- $1 \leq a \leq b \leq 100$
- $1 \leq x_i \leq 100$ for all $1 \leq i \leq n$

Sample Data

Input

```
3
6 1 5
1 1 1 1 1 1

5 2 10
4 7 1 9 3

3 2 5
4 8 5
```

Output

```
Case #1: 8
Case #2: 90
Case #3: impossible
```

SS15N11B Pick-up Sticks

Author: Christian Müller

Once again, Lea invited Bea over for a nice laid-back afternoon of playing games. Bea almost always lost to Lea in the games they played, so Lea was looking forward to a relaxed time.

However, Bea is sick of losing. She spent the last week practicing all the games she ever played against Lea and as soon as she arrives at Lea's house, she announces that she challenges Lea for the title of "Queen of Games". Intrigued, Lea does not back down from the challenge but announces that she will not hold back and will try to win every single game. (The title does include some nice perks - Lea is looking forward to making Bea call her "Queen Lea, Crusher of Rooks, Bringer of Checkmates and Protector of the King" in front of all her other friends.)

Before moving on to the strategy games, they warm up by playing a game called "Pick-up Sticks" ("Mikado" in German). This is a game where a bunch of wooden sticks are dropped on the table where they end up in a jumbled pile. The player whose turn it is then tries to pick up one stick without moving any of the other sticks. For every stick he can take, he gets a set amount of points. If he fails and moves any other stick, his turn is over, the sticks are rejumbled and the next player can try.

Figure 2: Pick-up Sticks, from `http://a.pragprog.com/magazines/2010-06/images/iStock_000001225226Small__10avt3__.jpg`

Lea is (as usual) quite skilled at this game. As long as there is no other stick that lies directly on top of the stick she is trying to pick up, she will never fail. For a given pile of sticks, can you tell her how many points she will get?

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 consists of a line containing two integers n m. n is the number of sticks that are still in the game. m is the number of intersections of sticks, where one stick lies directly on top of another one. A line of n space-separated integers $p_1 \dots p_n$ follows where p_i is the point value of the i-th stick. m lines follow. The j-th line contains two integers a_j b_j and means that there is a point where stick a_j lies directly on top of b_j.

Output

For each test case, print a line containing "Case #i: p" where p is the maximum number of points Lea can get.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10000$
- $1 \leq m \leq 100000$
- $p_i \in \{2, 3, 5, 10, 20\}$ for all $1 \leq i \leq n$
- $1 \leq a_j \neq b_j \leq n$ for all $1 \leq j \leq m$
- If stick a_j lies on top of stick b_j, then b_j can not lie on top of a_j for all $1 \leq j \leq m$.

Sample Data

Input

```
2
5 4
2 3 3 5 20
1 2
2 5
4 5
5 3

5 4
2 3 3 5 20
1 2
2 5
5 3
3 2
```

Output

```
Case #1: 33
Case #2: 7
```

SS15N11C Students

Author: Chris Pinkau

There are some changes going on at the TUM (Thomas Underwood University Markistan). Students have been rebelling all over the campus: lecture halls have been barricaded, mensa food has been refused, chairs have been tipped over... Lea has been following the news about the situation in Markistan with great interest. When she found out what the actual reason behind all this ruckus is, she wanted to help because it seemed to her that the problem could be solved with ease: all German documents published by the TUM must no longer use the word "`student`", but rather "`studierender`". The same is true for "`studentin`"/"`studierende`" and "`studenten`"/"`studierende`". So, Lea now wants to write a short program to do this, but to save precious time and work, she decides to only replace the patterns: "`entin`" will be replaced by "`ierende`", "`enten`" by "`ierende`", and "`ent`" by "`ierender`". (In particular, this means that "`enten`" should not be replaced by "`ierenderen`", and "`entin`" should not be replaced by "`ierenderin`".) Please help Lea and the TUM so that all students may once again attend their courses peacefully.

Input

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

Each test case starts with an integer n, the number of lines of the text. n lines follow, containing the text.

Output

For each test case, output "Case #i:" where i is its number, starting at 1. Beginning in the next line, output the input text modified by replacing the above mentioned patterns.

Constraints

- $t = 1$
- $1 \leq n \leq 1000$
- The input text contains at most 100000 characters.
- The input text contains only the lowercase alpha-numeric characters "a" to "z" or "0" to "9", or ".", ",", ":", "!", "?", "-", a space, or a line break.

Sample Data

Input

```
1
3
ich bin ein student
a potent element went through a vent
ein zentner enten enthaelt keine studenten
```

Output

```
Case #1:
ich bin ein studierender
a potierender elemierender wierender through a vierender
ein zierenderner ierende ierenderhaelt keine studierende
```

SS15N11D Portable Device

Author: Philipp Hoffmann

Lea, always equipped with the latest technology, has sniped something magical off of ebay: A portable device that allows you to walk through walls. Using complicated quantum effects, you can lean against a wall (which may be at most 70 cm thick), activate it and you will simply emerge on the other side.

Of course this comes at a huge cost: The device can only be used three times until its batteries are dead. Lea wants to use this device in the upcoming "Competition of Mazes", a contest to see who can escape first from a given maze. She just needs to plan her route perfectly.

Given the map of the maze, tell her which walls she needs to pass through to reach the target as fast as possible. Even though you pass through a wall and thus move two grid positions, using the device counts as one step. Note that Lea can only pass through a wall if the location directly behind it is walkable. Furthermore, Lea cannot walk diagonally. Since she does not want to be identified as a cheater, she has to exit the maze through the usual exit, not by using the portal device on an outer wall.

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 w and h, the width and height of the maze. h lines follow describing the maze, with w characters each.

Each line will contain only the characters "@", "_" and "$*$" , where "@" is a wall (of thickness 70 cm), "_" is a walkable position and "$*$" is the position where Lea starts.

Output

For each test case, print a line containing "Case #i:" where i is its number, starting at 1. Print three more lines containing either two coordinates x y, the (1-based) positions of walls Lea should skip to reach the target as fast as possible, or "unused" if you did not use that charge of the device. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $3 \leq w, h \leq 200$
- The maze will always have a unique exit, i.e. a walkable position on the border of the grid.
- The exit will always be reachable from the starting position, even without the portable device.

Sample Data

Input

```
3
5 5
@@@@@
@__*@
@_@@@
@__@@
@@_@@

7 7
@@@@@@@
______@
@@@@@_@
@_____@
@_@@@@@
@___*_@
@@@@@@@

10 10
@@@@@@@@@@
@@@______@
@@@@@_@_@@
@_@@___@_@
__@@@_@__@
@_@_____@@
@___@@__@@
@_@@_@@@*@
@________@
@@@@@@@@@@
```

Output

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

SS15N11E Tetris

Author: Stefan Toman

Just one more level! Lea is on her way to finally beat Bea's highscore at Tetris. They play an advanced Tetris game with unusual types of pieces which is extremely hard to master. The pieces are a connected set of tiles which are of size 1×1. The pieces fall down from top of the screen until they hit another piece. The game is so fast that is is impossible to position pieces while they are falling down. Lea can just move and rotate them in the fraction of a second before they enter the board. Formally speaking, Lea can choose a horizontal offset and a rotation for her tile, then it is falling down until it shares a horizontal edge with another piece or the bottom of the board. The final position of the piece is legal if and only if the piece does not intersect with tiles already on the board and fits completely on the board. If a piece does not fit on the board the game is over: Lea keeps her score up to this piece.

If one line is fully filled with tiles it is erased and all tiles above fall down exactly one line. To make advanced strategic planning possible, the game shows the next few pieces to the player. Can you tell Lea how many lines she can fill up completely?

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 an integer n, the number of pieces that will be falling down. $n + 1$ blocks follow, the first one describing the board and the next n ones describing the pieces that will arrive in order.

Each block starts with two integers w and h, the block's width and height. h lines follow, containing w characters each. Each of these characters will be X for a used tile or a dot for an unused tile.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is the maximum number of lines that Lea can eliminate.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n \leq 4$
- $5 \leq w \leq 10$, $5 \leq h \leq 20$ for the board.
- $1 \leq w \leq 5$, $1 \leq h \leq 5$ for the pieces.

- All pieces will be connected and there will not be empty lines (horizontal or vertical) in the pieces falling down

- Initially, the board does not contain lines full of tiles.

Sample Data

Input

```
3
1
5 5
.....
X....
XX...
XX...
XXX..
4 1
XXXX

1
5 5
.....
.....
.....
X...X
X.X.X
2 3
XX
.X
XX

3
7 10
.......
.......
.......
.......
X......
XX..XXX
XXX.XXX
XXX.XXX
XXXXXX.
.XXX.XX
3 2
XX.
.XX
2 3
.X
.X
XX
2 2
XX
XX
```

Output

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

SS15N11F Subway Stations

Author: Chris Pinkau

It was only quite recently that Lea has moved to her new apartment. She is still getting used to the local public transportation system and all its different stations and lines. And you cannot believe how vastly big and complex the local transportation network is. What brings it from bad to worse is the fact that there are ever ongoing constructions and constant changes in it. This is why most people just get on the subway on the next station they can find and hop off on another randomly, doing this for quite some time until they are close enough to their destination to walk the remaining distance. But this is nothing for Lea, she wants to plan her trips with the subway exactly and find out the minimal number of transfers between subway lines to get to wherever she wants. Many have tried this before her, as many have failed. But this does not bring Lea down. Even more so, she wants to succeed. And with your help, what can go wrong?

Input

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

Each test case begins with a line containing an integer n, the number of subway stations, and four floating point numbers x_{start} y_{start} x_{end} y_{end}, describing the coordinates of Lea's starting and destination stations, respectively. n lines follow, each containing an integer m_i and $2 \cdot m_i$ more integers $x_{i,1}$ $y_{i,1}$ $\ldots$ x_{i,m_i} y_{i,m_i}, describing the i-th subway line that starts with a station at $(x_{i,1}, y_{i,1})$ and goes in a straight line on to $(x_{i,2}, y_{i,2})$, then straight to $(x_{i,3}, y_{i,3})$, and so on until the terminal station at (x_{i,m_i}, y_{i,m_i}).

Output

For each test case, output "Case #i: x" where i is its number, starting at 1, and x is either the minimal number of subway line transfers required to get from (x_{start}, y_{start}) to (x_{end}, y_{end}); or "impossible" if there is no way to travel from start to end. The number of stations travelled does not matter, only the number of transfers.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq m_i \leq 100$ for all $1 \leq i \leq n$
- $0 \leq x_{i,j}, y_{i,j} \leq 10000$ for all $1 \leq i \leq n$ and $1 \leq j \leq m_i$
- Each point $(x_{i,j}, y_{i,j})$ is a station, for all $1 \leq i \leq n$ and $1 \leq j \leq m_i$.
- There is a station at each intersection point of two subway lines.

- (x_{start}, y_{start}) and (x_{end}, y_{end}) are both stations in the network, i.e., each of them either appears in the input, or is an intersection point.
- $x_{start}, y_{start}, x_{end}$, and y_{end} are given with up to 15 digits after the decimal point.

Sample Data

Input

```
3
3 1.0 3.0 0.0 4.0
4 1 6 4 6 4 2 1 0
3 0 4 6 4 8 2
2 4 0 1 3

3 2.0 0.0 6.0 6.0
4 2 0 2 6 4 2 4 6
3 1 4 6 4 6 8
5 3 8 5 7 7 5 3 3 1 4

3 2.0 0.0 3.0 2.0
4 2 0 2 5 6 5 6 0
2 1 3 7 3
2 3 2 5 2
```

Output

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

SS15N11G Pizza Party

Author: Philipp Hoffmann

Lea has decided that it is time to host another party! She invited all of her friends and ordered a huge amount of pizza. When the pizza boy brought the food, she was surprised: The pizza was an equilateral triangle!

Making the best out of the situation, she decided to cut the pizza into equally sized equilateral triangles, one for each party attendee. Unsure whether this is possible without leftovers, she asked you to do the calculations. In the event that it is not possible, tell her how many more friends she has to invite over to finish the pizza completely.

Input

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

Each test case consists of a single line containing an integer n, the number of people at the party.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is the number of people she has to invite additionally to those already present. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 50$
- $1 \leq n \leq 10^{18}$

Sample Data

Input

```
3
2
4
7
```

Output

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

SS15N11H Jetpack Jumps

Author: Christian Müller

Recently, Lea found a new game on the app store of her choice - "Jetpack Jumps". Since then, she spent countless hours jumping from platform to platform with her jetpack.

It works like this: Lea starts on the leftmost of an infinite series of platforms. She now jumps to the right one platform at a time, trying to get as far as possible (and break all the highscores) without falling into the bottomless pit between the platforms.

The distance between the platforms always increases by 1 meter, i.e. the second platform comes 1 meter after the first, the third comes 2 meters after the second, and so on. Lea herself can only jump a single meter, but can extend her jump by using her jetpack. With the jetpack, Lea can jump as far as she wants to. However, every meter that she uses the jetpack uses up fuel - the longer the jump, the more fuel she needs. Thus, for a jump of x meters, she needs $(x-1)^2$ liters of fuel (since she has to accelerate upwards first).

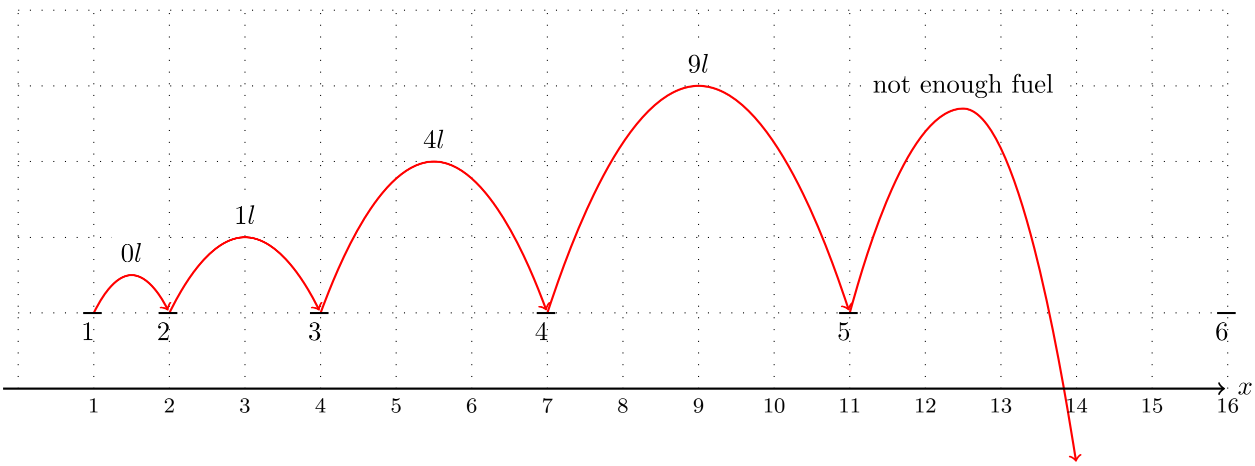

Figure 3: Illustration of the sample input, case 2.

Lea starts with a set amount of fuel. Assuming she plays perfectly, can you tell her which platform she can reach before she eventually runs out of fuel and tumbles down into the darkness (or has to buy the DLC for additional fuel)?

Input

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

Each test case consists of a single line containing an integer f, the amount of fuel Lea starts with (in liters).

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1 and x is the index of the platform Lea can reach (Lea starts on platform 1). Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$.
- $1 \leq f \leq 10^{24}$

Sample Data

Input

```
4
5
15
35
100
```

Output

```
Case #1: 4
Case #2: 5
Case #3: 6
Case #4: 8
```

SS15N11I Snowball Fight

Author: Christian Müller

Winter is a special time. The first real snowstorm hits, the lakes freeze over, the houses become snowcapped, trees have finally shed all their colors and become frosty skeletons. Everything outside becomes quiet.

Well, not entirely... Ever since they were small children, Lea and her friends have always loved this time. Why? Because once a year, the whole town comes together and celebrates the "Snowball Arena: Free-for-all". For a day, the whole town stands still and anyone who is spotted on the streets can be subject to snowball bombardment. It is an event of joy and of tears, where grown men cry, bombarded by endless blizzards of snowballs, thrown by the hands of children. To make it even more fun, the townspeople have dug trenches on the central square so there are now several fronts that can be besieged.

Of course, Lea takes part in all that. Right now, she is lying alone in one of the trenches and is bombarded by one particularly persistent fellow. To avoid being hit, she needs to keep her head low - thus, she cannot spot her attacker directly. All his snowballs came from the same direction however, so to retaliate, she only needs to know how far her attacker is away.

Luckily, she notices something - to the side of the central square, there is a huge new building[1]. There are no visible windows on the front wall, but rather the whole wall of that building is a huge mirror. Noticing that this could give her a substantial combat advantage, she watches out for movements in the mirror. As soon as she spots someone, she wants to lob a snowball over the siege lines and hit that person right in the face. Can you help her with the target computations?

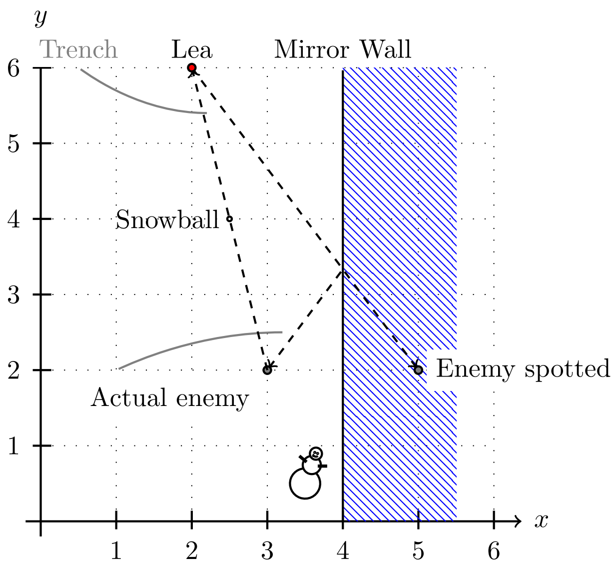

Figure 4: Illustration of the sample input, case 1.

[1]It is owned by some huge tech corporation. Lea forgot the name, but it was something like "Mirror's Ledge".

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 consists of 4 lines. The first line contains the integers x_{Lea} y_{Lea}, Lea's coordinates. The second line contains the doubles x_{Snow} y_{Snow}, some in-flight coordinates of one of the snowballs that came from the direction of the attacker. The third line contains the integers x_{Wall1} y_{Wall1} x_{Wall2} y_{Wall2}, the coordinates of the wall. The fourth line contains the doubles x_{enemy} y_{enemy}, the projected coordinates of the enemy she saw in the mirror.

Output

For each test case, print a line containing "Case #i: x y!" where i is its number, starting at 1, and (x, y) are the coordinates of the enemy. Your solution is considered correct if the area is accurate to four decimal places. Each line of the output should end with a line break.

Constraints

- $1 \le t \le 20$
- $y_{Wall1} = 0$
- $y_{Wall2} = 30$
- All given coordinates are between 0 and 30.
- All points are at least 10^{-4} apart.
- All points are at least 10^{-4} away from the wall.

Sample Data

Input

```
1
2 6
2.5 4.0
4 0 4 30
5 2
```

Output

```
Case #1: 3.0 2.0
```

SS15N11J Scout the Castle

Author: Philipp Hoffmann

When the weather cools and winter is around the corner, times are hard. Especially when there is war and a new king pops up at every occasion. Right now, King Davos is planning to take the capital for himself. He just needs to find the right angle of attack, lest he be trapped and burned alive or worse.

The capital is surrounded by water to the east and by land on the remaining three sides. King Davos, afraid of water, because you never know what lurks beneath, plans to attack the city from land. He and his army are north of the city walls right now. To get a better picture, he wants to send scouts around the city. However, he needs to be careful not to be discovered.

The city is surrounded by various villages and some impassable terrain like mountains. He has drawn a map of the city and the terrain around it. The scouts should now circle the city without getting too far away (otherwise they won't be able to scout anything). However, to not draw attention, at most one scout may pass through every village.

Since King Davos has never been good with staying undetected, he sends a raven with the map and his instructions to his advisors. By some mistake, the raven ends up with Lea who is instantly intrigued and tries to figure out the number of scouts King Davos can send.

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 three integers w h and n, the width w and height h of the map which is a grid, and the number of impassable locations n. All following coordinates are 1-based.

The next line contains four integers c_x c_y c_w c_h meaning that the city occupies a square starting at grid position (c_x, c_y) of width c_w and height c_h in the grid, extending to the right and downward.

n lines follow, describing the impassable locations. Each line contains four integers x_i y_i w_i h_i, describing an impassable square starting at grid position (x_i, y_i) of width w_i and height h_i in the grid, extending to the right and downward.

The scouts will sneak along the shore until they enter the grid from the east, above the city. Thy may leave the grid again on the east side, below the city. They can move horizontally or vertically between passable terrain, but may of course not enter the city. Each passable grid location corresponds to a village, i.e. may only be passed by at most one scout.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is the maximal number of scouts that King Davos can send without being detected. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $2 \leq w \leq 10000$
- $3 \leq h \leq 10000$
- $0 \leq n \leq 100$
- $1 \leq c_x, c_y \leq 51$
- $c_x < c_x + c_w = w + 1$
- $c_y < c_y + c_h < h + 1$
- $1 \leq h - (c_y + c_h - 1) \leq 50$
- $1 \leq x_i < x_i + w_i \leq w + 1$ for all $1 \leq i \leq n$.
- $1 \leq y_i < y_i + h_i \leq h + 1$ for all $1 \leq i \leq n$.
- An impassable location will never overlap with the city or another impassable location.

Sample Data

Input

```
3
5 6 1
3 4 3 1
1 1 1 1

4 6 4
3 3 2 2
3 2 2 1
2 3 1 1
1 5 1 1
3 6 2 1

8 10 5
8 3 1 6
2 2 5 1
2 3 1 3
6 3 1 1
4 4 1 1
4 5 4 1
```

Output

```
Case #1: 2
Case #2: 1
Case #3: 2
```