Algorithms for Programming Contests
SS20 - Week 13

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Mikhail Raskin, Tobias Meggendorfer, Stefan Jaax, A.R.Balasubramanian,
Christoph Welzel

Welcome to our practical course! This problem set is due by

Tuesday, 21.07.2020, 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:

You may work together in teams of two students.

Six points are awarded for each problem, resulting in 60 points in total. However, only 40 of them will count towards the total number of points, and the additional 20 points will be counted as bonus points.

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
Flowery Trails

Figure A.1: `http://covermyfb.com/media/covers/thumb/7354-flower-trail.jpg`

In order to attract more visitors, the manager of a national park had the idea of planting flowers along both sides of the popular trails, which are the trails used by common people. Common people only go from the park entrance to its highest peak, where views are breathtaking, by a shortest path. So, he wants to know how many metres of flowers are needed to materialize his idea. For instance, in the park whose map is depicted in the figure, common people make only one of the three following paths (which are the shortest paths from the entrance to the highest peak).

- At the entrance, some start in the rightmost trail to reach the point of interest 3 (after 100 metres), follow directly to point 7 (200 metres) and then pick the direct trail to the highest peak (620 metres).
- The others go to the left at the entrance and reach point 1 (after 580 metres). Then, they take one of the two trails that lead to point 4 (both have 90 metres). At point 4, they follow the direct trail to the highest peak (250 metres).

Notice that popular trails (i.e., the trails followed by common people) are highlighted in yellow. Since the sum of their lengths is 1930 metres, the extent of flowers needed to cover both sides of the popular trails is 3860 metres ($3860 = 2 \cdot 1930$).

Given a description of the park, with its points of interest and (two-way) trails, the goal is to find out the extent of flowers needed to cover both sides of the popular trails. It is guaranteed that, for the given inputs, there is some path from the park entrance to the highest peak.

Input

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

Each test case begins with a line containing two integers P and T. P is the number of points of interest and T is the number of trails. Points are identified by integers, ranging from 0 to $P - 1$. The entrance point is 0 and the highest peak is point $P - 1$. T lines follow, each characterising a different trail. The i-th line contains three integers, p_{i1}, p_{i2}, and ℓ_i, which indicate that the (two-way) trail links directly points p_{i1} and p_{i2} (not necessarily distinct) and has length ℓ_i (in metres).

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the extent of flowers (in metres) needed to cover both sides of the popular trails.

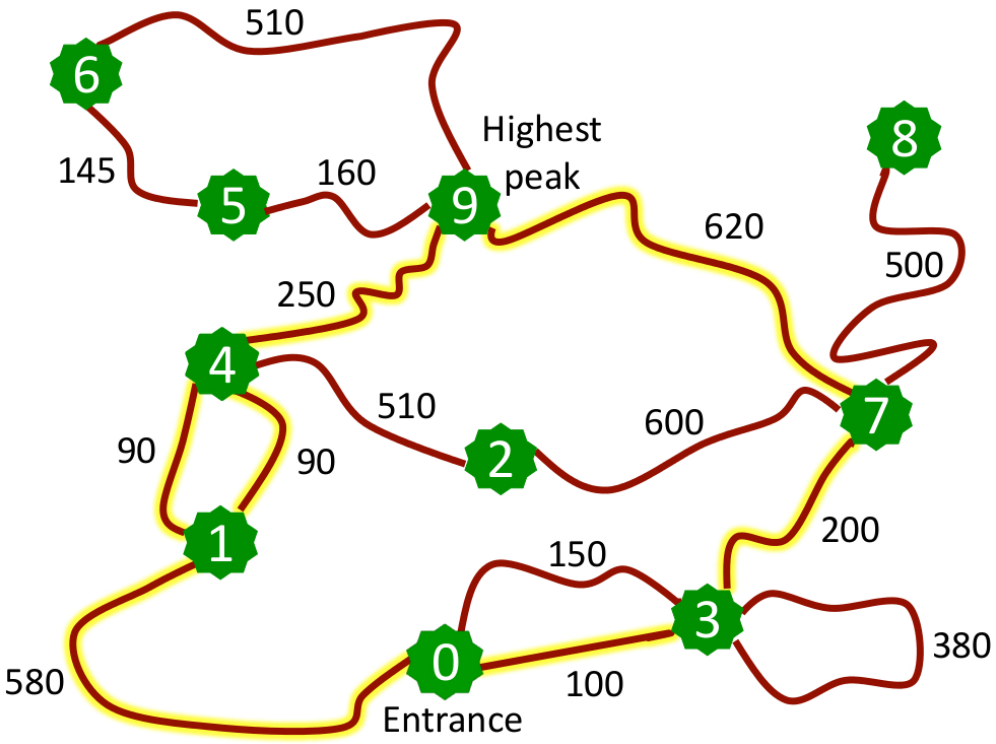


Figure A.2: Illustration of the sample input, Case #1, with the shortest paths marked in yellow.

Constraints

- $1 \leq t \leq 2$
- $2 \leq P \leq 10000$
- $1 \leq T \leq 250000$
- $0 \leq p_{i1}, p_{i2} \leq P - 1$ for all $1 \leq i \leq T$
- $1 \leq \ell_i \leq 1000$ for all $1 \leq i \leq T$

Sample Input 1

```
2
10 15
0 1 580
1 4 90
1 4 90
4 9 250
4 2 510
2 7 600
7 3 200
3 3 380
3 0 150
0 3 100
7 8 500
7 9 620
9 6 510
6 5 145
5 9 160

4 7
0 1 1
0 2 2
0 3 10
0 3 3
1 3 2
2 3 1
1 1 1
```

Sample Output 1

```
Case #1: 3860
Case #2: 18
```

This page is intentionally left blank.

Problem B
Redundancy

After a long period of job hunting, Lea decides she wants to become a city planner. Her boss, Robert Moses, is known for his extravagant approach to city planning. He challenges Lea with the following task:

> I want you to restructure our subway network, but it better be a perfectly efficient part of our perfect city plan. I have a strong aversion against redundancy, and it's probably in your best interest that I don't feel like you are a redundant employee. So, here's a list of all stations... (*pulls out a list of stations*) and here's a map of all existing connections... these connections are in need of renovation, and that is costly, here is a table of the estimated renovation cost per connection. Some connections are in such bad condition that it is probably best to demolish them and build new ones. Here's an estimate of what it costs to demolish a connection. And here is an estimate of what it costs to build a new one. Now, here's the deal: You may decide if you like to renovate an existing connection or demolish it. You may also build new connection, but, as I said, I hate redundancy (already hating myself for repeating this statement), so in the end I want our network to be connected, without superfluous connections. And you should keep the total cost of demolishing, renovating, and building new connections to a minimum.

Can you help Lea demonstrate that she is not a redundant employee?

Input

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

The first line of each test case contains two space-separated positive integers n and e. The value n denotes the number of subway stations numbered $1, \ldots, n$, and e denotes the number of existing links between two stations. Then e lines follow. Each of these lines consists of four space-separated positive integers v_1, v_2, d, r, where $1 \leq v_1, v_2 \leq n$, $v_1 \neq v_2$, establishing an existing connection between stations v_1 and v_2, with renovation cost r and cost and demolition cost d. Finally $n \cdot (n-1)/2$ lines follow, each containing three space-separated integers v_1, v_2, c, denoting the cost of contructing a link between v_1 and v_2 for all pairs such that $v_1 < v_2$. All connections are assumed to be symmetric: a connection between v_1 and v_2 also serves as connection between v_2 and v_1.

Output

For each test case, output one line containing "Case #i: y", where i is the number of the case and y is the total minimal cost incurred by demolitions of existing connections, renovation of connections that are not demolished, and construction of new connections – subject to the constraint that the final network shall be connected and there shall be no connection that can be removed without breaking connectivity.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $0 \leq e \leq n \cdot (n-1)/2$
- $0 \leq d, r, c \leq 100$

Sample Input 1

```
1
4 5
1 3 8 9
1 2 7 20
2 4 5 7
2 3 3 7
3 4 7 1
1 2 4
1 3 6
1 4 7
2 3 3
2 4 2
3 4 6
```

Sample Output 1

```
Case #1: 27
```

Problem C
Partytime

Lea's friend Bea wants to throw a huge birthday party. Of course, birthday parties need music. Since Lea is known for her DJ skills, Bea asks her to compile a nice playlist. However, Bea has some additional constraints. Firstly, all tracks have to be cross-faded for 5-10 seconds to have smooth transitions. This means that 5-10 seconds before the end of the current track, the next track already starts playing, slowly ramping up in volume. Secondly, for dramatic purposes, Bea wants to have cross-fades happening at specified points in time, i.e. a cross-fade should be in progress at all of the given timestamps. Bea is already happy if the fade is starting or ending at that exact point in time. Moreover, Lea is a perfectionist and refuses to transition wildly between all songs in her library. Each song belongs to a different genre and one track may only follow another if they belong to similar genres. Due to technical restrictions, transitions can only start and end at full seconds, e.g. fading from track A to B cannot start at 60.5 seconds. Can you help Lea compile a playlist satisfying the requirements?

Input

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

Each test case starts with a single comprising the number of genres g, the number of songs n, and the number of cross-fade timestamps m. Then, g lines follow. Each line starts with a number k_i, the amount of genres compatible with genre i, followed by k_i numbers, denoting the set of compatible genres. This compatibility relation is guaranteed to be symmetric. Note that some genres potentially are not compatible with themselves. Then, n more lines follow, each comprising two numbers l_i and g_i, denoting the length l_i in seconds and genre g_i of song i. Finally, a last line follows, containing m timestamps c_i, indicating where a cross-fade should happen.

N.B.: The first track does not have a "fade-in", time starts at 0 seconds, and the cross-fade constraints are also satisfied if the fade starts or ends precisely at the given timestamp.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is either `yes` if a playlist satisfying the constraints exists or `no` if no such playlist exists.

Constraints

- $1 \leq t \leq 20$
- $1 \leq g \leq 10$
- $1 \leq n \leq 50$
- $1 \leq m \leq 50$
- $0 \leq k_i \leq g$
- $20 \leq l_i \leq 2{,}000$
- $0 \leq c_i \leq 10{,}000$

Sample Explanation

In the first sample, Lea can play song 1 from 0 to 20 seconds and fade to song 2 at 10. Then, we have at time 10. The second song would end at 40 seconds. Fading to track 1 at 30 seconds and back to track 2 at 45 gives the desired result. In the second sample, we clearly have to start with track 1 and start fading with 10 seconds at 50. The only possible next track would be the track 2 (due to genre restrictions), however we cannot have a fade starting at 110 already, since track 2 would then be playing until 150.

Sample Input 1

```
1
2 2 2
1 2
1 1
20 1
30 2
10 50
```

Sample Output 1

```
Case #1: yes
```

Sample Input 2

```
1
3 3 2
2 2 3
2 3 1
2 1 2
60 1
100 2
80 3
50 110
```

Sample Output 2

```
Case #1: no
```

Problem D
Absurd Prices

Surely you know that supermarkets, shopping centres, and indeed all kind of vendors seem to have fallen in love with the digit 9, for that digit occurs most often in the price of a product, preferably at the least significant positions. Your favourite chocolate bar might cost 99 cents, just right to be able to advertise that it costs less than 1 euro. Your new bicycle might cost 499.98 euros, which, of course, is less than 500 euros.

While such comparisons are mathematically sound, they seem to impose a certain amount of stupidity on the customer. Moreover, who wants to carry home those annoying small coins you get back as change?

Fortunately, the FIFA has not adopted this weird pricing scheme: a ticket for the final in the first category for example costs 900 dollar, in the second category 600 dollar and in the third category 400 dollar. These prices may only be regarded weird for other reasons.

We want to distinguish between *absurd* prices like 99 cents, 499.98 euros, etc. and normal prices. To measure the absurdity of a positive integer, do the following:

- Eliminate all trailing zeros, i.e., those in the least significant positions, from the number. You now have a positive integer, say x, with a non-zero digit d at its end.
- Count the number of digits, say a, of the number x.
- if $d = 5$ the absurdity of the number is $2 \cdot a - 1$
- otherwise, the absurdity of the number is $2 \cdot a$

For example, the absurdity of 350 is 3 and the absurdity of 900900 is 8. Using the measure of absurdity, we can define what we call an absurd price: A price c is absurd if and only if the closed interval $[0.95 \cdot c, 1.05 \cdot c]$ contains an integer e such that the absurdity of e is less than the absurdity of c.

Given a price in cents, go ahead and tell whether it is absurd!

Input

The first line of the input consists of the number t of test cases to follow. Each test case is specified by one line containing an integer c.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is "`absurd`" or "`not absurd`". Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 3000$
- $1 \leq c \leq 10^9$

Sample Input 1	Sample Output 1
4 99 49998 90000 970000000	Case #1: absurd Case #2: absurd Case #3: not absurd Case #4: absurd

This page is intentionally left blank.

Problem E
Templonian Maps

During her vacation in Templonia, Lea plans a sailing trip across their six seas (the seventh recently evaporated due to global warming). Global warming is changing the shape of all other islands, too. Since the maps of some far away islands still have not been updated, Lea plans to additionally take care of this issue while she is on vacation. On each island she needs a location to set up a theodolite in order to precisely measure the changes. Such a location needs to be (i) above water and (ii) flat. Luckily, templonian islands are very smooth and described by mathematical functions. Lea of course remembers all these function from her last visit. Given that, can you help her find a good spot to set up her theodolite?

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 single line, starting with a number k, the number of symbols in the function. This number is followed by the function describing the island, given in reverse polish notation (`https://xkcd.com/645/`). The functions contain the variables `x` and `y`, floating point numbers, basic binary arithmetic symbols `+`, `-`, and `*`, and the functions `NEG`, `SIN`, and `COS`, denoting unary negation, sine, and cosine, respectively. The symbols are separated by spaces. For example, the function $2 \cdot x + \sin(y)$ would be written as `2 x * y SIN +`.

Output

For each test case, output a line containing "Case #i: x, y", with x, y being a suitable spot for the theodolite. In particular, for the given function f, it is required that (i) $f(x, y)$ is bigger than zero and (ii) f has a gradient with infinity norm less than 10^{-3} at x, y.

Constraints

- $1 \leq k \leq 50$
- $f(x, y) < 0$ for $||(x, y)||_2 = 10{,}000$
- f is defined on the whole of $\mathbb{R}^2$
- $f(0, 0) > 0$

Sample Input 1	Sample Output 1
`1` `17 2 x 0.5 - x 0.5 - * y 0.5 - y 0.5 - * + -`	`Case #1: 0.5 0.5`

This page is intentionally left blank.

Problem F
Box Union

Given some axis-aligned rectangles, compute the area of their union.

Input

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

The first line of each case contains an integer n, the number of rectangles. n lines follow, each containing four integers $x_1\ y_1\ x_2\ y_2$. (x_1, y_1) is the lower left corner of the a rectangle, (x_2, y_2) the upper right corner.

Output

For each test case, print a line containing "Case #i: x", where x is the area of the union of the given rectangles.

Constraints

- $1 \leq n \leq 100$
- $0 \leq x_1 < x_2 \leq 1000$
- $0 \leq y_1 < y_2 \leq 1000$

Sample Input 1	Sample Output 1
2 2 1 1 3 3 2 2 4 4 3 1 1 3 2 0 0 5 5 1 6 3 7	Case #1: 7 Case #2: 27

This page is intentionally left blank.

Problem G
An Industrial Spy

Industrial spying is very common for modern research labs. I am such an industrial spy – don't tell anybody! My recent job was to steal the latest inventions from a famous math research lab. It was hard to obtain some of their results but I got their waste out of a document shredder. I have already reconstructed that their research topic is fast factorization. But the remaining paper snippets only have single digits on it and I cannot imagine what they are for. Could it be that those digits form prime numbers? Please help me to find out how many prime numbers can be formed using the given digits.

Input

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

Each test case consists of a single line. This line contains one integer n, and a sequence m consisting of n digits that are on the paper snippets.

Output

For each test case, print one line containing "Case #i: x" where i is its number, starting at 1, and x is the number of different primes that can be reconstructed by shuffling the digits of m. You may ignore digits while reconstructing the primes (e.g., if you get the digits 7 and 1, you can reconstruct three primes 7, 17, and 71). Reconstructed numbers that (regarded as strings) differ just by leading zeros are considered identical (see the fourth case of the sample input).

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 7$
- $0 \leq m_i \leq 9$ for all $1 \leq i \leq n$

Sample Input 1	Sample Output 1
4 2 17 7 1276543 7 9999999 3 011	Case #1: 3 Case #2: 1336 Case #3: 0 Case #4: 2

This page is intentionally left blank.

Problem H
Rival Dice

Lea's friends, Bea and Kea each have a skewed dice with the large numbers of sides for an exciting new game. The points count in this game can be negative, and can be the same on different sides. Also it is possible that some sides carry points, but never come up however you throw the die. Bea and Kea know the probabilities of each outcome on each die, and argue which die is better.

Lea wants to settle the argument by entangling the dice by a magic enchantment. The enchantment sets a probability for each pair of outcomes on these dice such that one die always rolls at least as high as the other one. However, the total probabilities for each outcome for each single die must not be changed. Lea wonders if she will succeed at that task. Surely such an enchantment is not beyond her skills, but only as long as it does not contradicts the laws of mathematics! Check whether it is possible to choose the probabilities in such a way.

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, m, and d. Here n and m are the numbers of sides on Bea's and Kea's dice, and d is the denominator used for probabilities.

Two lines follow with n space-separated integers each. The former line contains p_i, the numerators of probabilities of sides for Bea's die. The latter line contains x_i, the point values of the sides. Note that two sides might have the same amount of points.

Two more lines follow with m space-separated positive integers each, describing Kea'sdie in the same way.

Output

For each test case, output one line containing "Case #i: Bea" if the first die can be proven better by an enchantment, "Case #i: Kea" if the second die can, and "Case #i: depends" otherwise. In the case if the two dice have the same distribution of points, output "Case#i: balance".

Constraints

- $1 \leq t \leq 10$
- $1 \leq m, n \leq 500$
- $0 \leq d \leq 1000000000$
- $p_1 + \ldots + p_n = q_1 + \ldots + q_m = d$
- $-1000000000 \leq x_i, y_i \leq 1000000000$
- $0 \leq p_i, q_i \leq 1000000000$

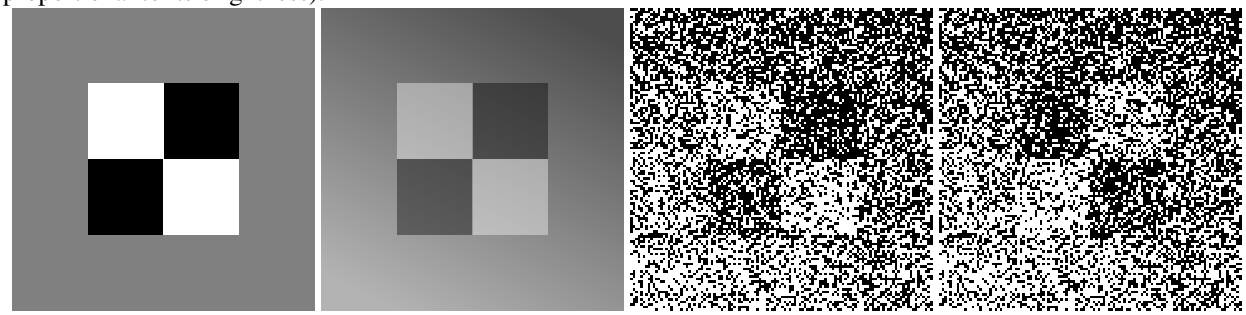
Sample explanation

In the first case the first die rolls 3 with probability $5/6$. However Lea enchants the dice, both pairs $(3, 2)$ and $(3, 4)$ will appear. Thus the answer is "depends". In the second case Lea can enchant the dice to roll $(1, 1)$ with probability $1/6$, $(3, 1)$ with probability $3/6$, $(3, 2)$ with probability $2/6$. Thus the answer is "Bea".

Sample Input 1	Sample Output 1
2 2 2 6 1 5 1 3 4 2 2 4 2 2 6 1 5 1 3 4 2 1 2	Case #1: depends Case #2: Bea

Problem I
Seeing is Believing

Lea has taken some perfectly aligned (but not always centered) pictures of a checkerboard! Although a very small checkerboard (2×2 cells). And under imperfect light (creating a linear gradient of gray with a random direction and unknown intensity, with the checkerboard cells being brighter or darker by the same amount — the gradient still applies). And in black and white — literally black and white, not grayscale (the probability of a pixel to be white is proportional to its brightness).



Still, it is visible whether the two black cells are in the bottom left and top right or bottom right and top left. Lea wants to classify the pictures according to the orientation of the black diagonal. Can you help her?

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 space-separated integers n and m. n lines of length m each follow. In each line, the character `x` denotes a black cell and the character `.` denotes a white cell; no other characters are present.

Output

For each test case output a line containing "Case #i: v", where v is either `\` or `/`, according to the orientation of the black diagonal.

Constraints

- $1 \leq t \leq 10$
- $1 \leq m \leq 1920$
- $1 \leq n \leq 1080$
- the side of the "overlay" squares is at least 15 cells (i.e. the minimum size of the "checkerboard" is 30×30)
- the cell changes the brightness/probability of white/black pixel by at least 15% compared to the underlying gradient

Sample Input 1	Sample Output 1
1 40 37 ...xx...x..xxx..xxxx.x.x.xx.xxxxx.... xx.x.x.....xxx.xx..x..xx.x.x.x.xxx.x xxx.........x.....xxx..xxxxxxxxxx.. x..x..........xx.xxxx.xxxxxxxxxx.xx .x.x.x.....x....xx.xx.xxx.xx.xx.xx .x.x.............x.xx.xxxx.xxxxxxxx ..xx.x..........xxxxxxxxxxxxxxx.xx .x.x...x.....xx...xxxxxx.xxx..xxx.x ..x.............xxxxxxxxxxx..x.. x...........x....xxx.xxxx.xxxxx.xx .x..x.....x....x..xx.xx.xxxxxxxx..x .x.x..x.....xx..x..xxxxx.xxxxx..xx.. x....x.........x...xxxxxxxxxxx.x.x. x.x.............x.xxxxx.xxx.xxx..x x..x.......x.x...xxxxx.xx.xx.xxxxxxx .x............xx.x.xxx.x.xx..xxxxxx x.....x.xx.....x.xxxxxx.xx.x... .xx.xxx..xxxx.x.xxx...x.....xx...x. .x..xxxxx.x.xxxx.xx.x....x.....x x...x.x...xxxxxxxxx.........x..xxx x...xxx.xx.xx..xxxxx..x.........x.. x.....x.xx..xxx.xx.....x.x...x.x...x. x.x.xx.xxx..xx..xxx.x...x.....x.... x...xx...xxxxxx..xx.............x.x x..xxx.xx.xxx...x.x.....x......... .x.xxxx..x...xxx.xx.........x...xx.. x.xxx.x..xx.xx.xxx.........x...x.x xx...xxxx..x.xx.xxx...x.x.....xxx ...x.x..xx.xxxxx.xx.....x..x...x.... x.xxxx..x.xxxxx.xx..x....x...x....xx xx.xxx.xxxxxxx.xx.....x..x.....x.x ....x.xxx..xxx....x.....x.....xxx. xx....x.....x.x...xxx...x..x..xx. .xxxxx.....xx.xxxx.....x.x..xx. ..x...x.x.xx.x.xx.x.x.....x....xx..xx .....x...x..xx.xx..xxx.x.x..x.....x ....x..xx....xx....xx.....xxxx ..xxx..xx...xx.xx.....xx.....x.x... x.x....x.....xx..xx....x.....x.x.x ....x.x..xx.x..x.x.x.x....xxxx.x....x	Case #1: /

Problem J
Grachten

Damn! I did not only oversleep (and today is *the* contest day!) but I also got stuck somewhere in Delft on my way from the hotel to the contest site. Everywhere around me are grachten, these city-canals that are part of many cities in the Netherlands. I am in a bit of hurry, because the NWERC contest starts in a few minutes.

To make matters even worse, some bridges in Delft are closed due to a cycling race through the city. Thus, I decided to jump over some of the grachten instead of searching for open bridges.

Everyone knows that computer scientists like me are good at algorithms but not very good athletes. Besides, I am a bit faint-hearted and don't want to get wet. So I need your help to calculate the distance I have to jump over a gracht.

Luckily, I did attend the excursion in Delft city center yesterday, where I learned that all paving stones in Delft are squares and have the same size. This way, I can do some measurements on my side of the gracht (my units are paving stones):

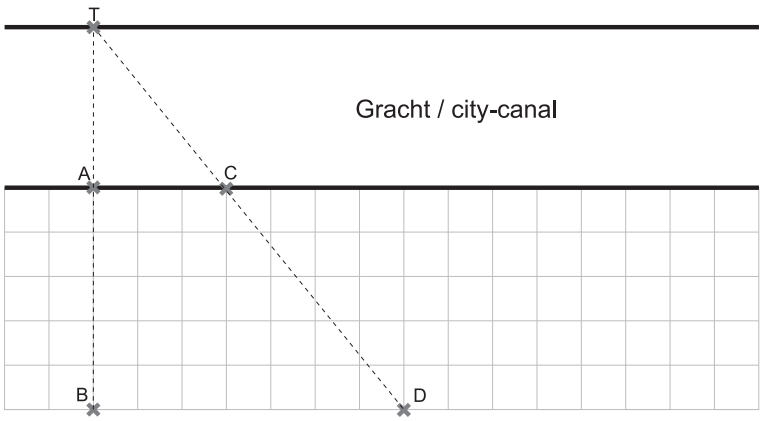


Figure J.1: Illustration of first sample input.

I walked from point C to point D via points A and B while counting the paving stones.

Points A and C are always on the edge of the gracht. Points B and D have the same distance to the gracht. The target point T is always on the edge of the other side of the canal; it is the intersection point of the line through B and A, and the line through D and C. The angle between AT and AC is 90 degrees, and the two edges of the canal are parallel lines.

Please calculate the distance between A and T (necessary jump distance) for me.

Input

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

For each test case, the input consists of one line containing three positive integers that specify the distances between A and B, A and C, and B and D.

Output

For each test case, print one line of output: "Case #i:" where i is its number, starting at 1, and the distance between A and T as a **reduced** fraction (i.e. remove all common factors of numerator and denominator).

Constraints

- $1 \leq t \leq 200$

- You may safely assume that no distance is larger than 1000 and the distance between B and D is larger than the distance between A and C.

Sample Input 1	**Sample Output 1**
6 5 3 7 5 3 8 1 2 3 23 42 47 500 500 1000 1 1 1000	Case #1: 15/4 Case #2: 3/1 Case #3: 2/1 Case #4: 966/5 Case #5: 500/1 Case #6: 1/999