The 2013 ACM ICPC
Asia Regional Contest
Dhaka Site
(Online Version)

Sponsored by IBM

Hosted by North South University
Dhaka, Bangladesh

30th November 2013
You get 20 Pages
11 Problems
&
300 Minutes

acm International Collegiate Programming Contest

IBM event sponsor

The 2013 ACM ICPC Asia Regional Contest Dhaka Site (Online Version) Sponsored by IBM

Hosted by North South University Dhaka, Bangladesh

30th November 2013 You get 20 Pages 11 Problems & 300 Minutes

Rules for ACM-ICPC 2013 Asia Regional Dhaka Site (Online Version):

a) Solutions to problems submitted for judging are called runs. Each run is judged as accepted or rejected by the judge, and the team is notified of the results.

b) Notification of accepted runs will **NOT** be suspended at the last one hour of the contest time to keep the final results secret. Notification of rejected runs will also continue until the end of the contest. But the teams will not be given any balloon and the public rank list will not be updated in the last one hour.

c) A contestant may submit a clarification request to judges. If the judges agree that an ambiguity or error exists, a clarification will be issued to all contestants.

d) While the contest is scheduled for a particular time length (five hours), the associate judging director has the authority to alter the duration of the contest in the event of unforeseen difficulties. Should the contest duration be altered, every attempt will be made to notify contestants in a timely and uniform manner.

e) **A team may be disqualified by the associate Judging Director** for any activity that jeopardizes the contest such as dislodging extension cords, unauthorized modification of contest materials, distracting behavior of communicating with other teams.

f) Eleven problems will be posed. So far as possible, problems will avoid dependence on detailed knowledge of a particular applications area or particular contest language. Of these problems at least two will be solvable by a first year computer science student, another two will be solvable by a second year computer science student and rest will determine the winner.

g) The decision of the judges is final.

h) If you want to assume that judge data is weaker than what is stated, then do it at your own risk :).

Rules for ACM-ICPC 2013 Asia Regional Dhaka Site (Online Version):

a) Solutions to problems submitted for judging are called runs. Each run is judged as accepted or rejected by the judge, and the team is notified of the results.

b) Notification of accepted runs will NOT be suspended at the last one hour of the contest time to keep the final results secret. Notification of rejected runs will also continue until the end of the contest. But the teams will not be given any balloon and the public rank list will not be updated in the last one hour.

c) A contestant may submit a clarification request to judges. If the judges agree that an ambiguity or error exists, a clarification will be issued to all contestants.

d) While the contest is scheduled for a particular time length (five hours), the associate judging director has the authority to alter the duration of the contest in the event of unforeseen difficulties. Should the contest duration be altered, every attempt will be made to notify contestants in a timely and uniform manner.

e) A team may be disqualified by the associate Judging Director for any activity that jeopardizes the contest such as dislodging extension cords, unauthorized modification of contest materials, distracting behavior of communicating with other teams.

f) Eleven problems will be posed. So far as possible, problems will avoid dependence on detailed knowledge of a particular applications area or particular contest language. Of these problems at least two will be solvable by a first year computer science student, another two will be solvable by a second year computer science student and rest will determine the winner.

g) The decision of the judges is final.

h) If you want to assume that judge data is weaker than what is stated, then do it at your own risk :).

A — Falling Ants

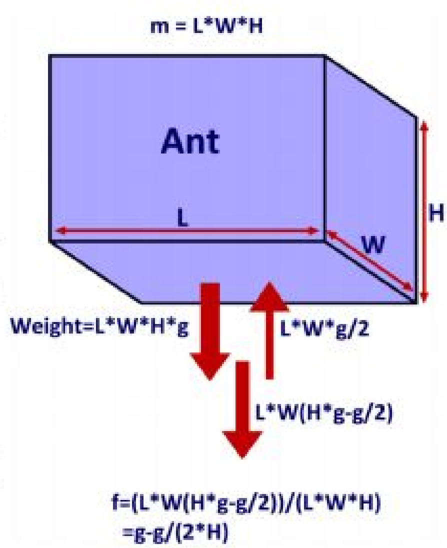
Input: Standard Input
Output: Standard Output

Ants make up 10% of the total world animal tissue. The total biomass of all the ants on Earth is roughly equal to the total biomass of all the people on Earth. However, unlike the people on Earth when they fall from a height they do not die for two reasons:

- They have so little mass relative to their air resistance that they fall slowly and, therefore, have little energy to dissipate when they hit the ground.
- Their bodies are tiny deformable tanks, well designed to absorb blows.

In general, small objects have less impact of gravitation on them because they have more surface area/volume compared to larger objects. For example consider a **(1x1x1)** cube. Its surface area is **6** and volume is **1**. So the ratio is **6:1** and for a **(10x10x10)** cube the surface area is **600** and volume is **1000**. So the ratio is **6:10**. Given the shape of many ants you will have to find out which ant has the highest effect of gravitation on it.

For simplicity we will assume the following things in this problem:



1. All ants are described as a box shaped object. A box shaped object is described with three integers **L**, **W**, and **H** which denotes the length, width and height of the object. So the volume of the ant is $(L \times W \times H)$.

2. The density (Mass per unit volume) is 1. So the mass of the above mentioned ant is also $(L \times W \times H)$ and so the weight is $(L \times W \times H) \times g$ (Here **g** is the acceleration caused by gravitation).

3. When an ant freely falls four sides are upright and so the faces at the top and bottom are parallel with the horizon. So the area of the plane facing bottom is always $L \times W$. For any ant the upward force put by the air is proportional to the area of the of the bottom face. To be specific it is $\frac{L \times W \times g}{2}$. After some manipulation it can be proved that the downward acceleration $f = g - \frac{g}{2H}$. So the downward acceleration actually depends solely on the value of **H** (as **g** is same for all ants).

Given the dimension of several ants, report the volume of the ant that has the highest downward acceleration. If there is a tie, report the one with the largest volume.

Ants make up 10% of the total world animal tissue. The total biomass of all the ants on Earth is roughly equal to the total biomass of all the people on Earth. However, unlike the people on Earth when they fall from a height they do not die for two reasons:

- They have so little mass relative to their air resistance that they fall slowly and, therefore, have little energy to dissipate when they hit the ground. - Their bodies are tiny deformable tanks, well designed to absorb blows.

In general, small objects have less impact of gravitation on them because they have more surface area/volume compared to larger objects. For example consider a (1x1x1) cube. Its surface area is 6 and volume is 1. So the ratio is 6:1 and for a (10x10x10) cube the surface area is 600 and volume is 1000. So the ratio is 6:10. Given the shape of many ants you will have to find out which ant has the highest effect of gravitation on it.

For simplicity we will assume the following things in this problem:

1. All ants are described as a box shaped object. A box shaped object is described with three integers L, W, and H which denotes the length, width and height of the object. So the volume of the ant is

(HWL ×

×) .

2. The density (Mass per unit volume) is 1. So the mass of the above mentioned ant is also

(HWL ×

×) and so the weight is

(

gHWL × ×) × (Here g is the acceleration caused by gravitation).

3. When an ant freely falls four sides are upright and so the faces at the top and bottom are parallel with the horizon. So the area of the plane facing bottom is always

WL×

. For any ant the upward force put by the air is proportional to the area of the of the bottom face. To be specific it is

gWL ×

× 2

. After some manipulation it can be proved that the downward acceleration

gf =

- 2

g H

.

So the downward acceleration actually depends solely on the value of H (as g is same for all ants).

Given the dimension of several ants, report the volume of the ant that has the highest downward acceleration. If there is a tie, report the one with the largest volume.

A Falling Ants

Input: Standard Input Output: Standard Output

Input

The input file contains at most **500** test cases. The description of each test case is given below:

First line of each test case contains an integer **T ($T \le 100$)** which denotes the total number of ants to consider. Each of the next **T** lines contains three integers which denote the value of **L**, **W** and **H ($1 \le L$, $W, H \le 50$)** of an ant respectively.

Input is terminated by a line containing a single zero.

Output

For each set of input produce one line of output. This line contains the volume of the ant that has the highest downward acceleration. If there is a tie report the ant with the highest volume.

Sample Input	Output for Sample Input
3 3 4 5 12 1 5 20 10 4 3 3 4 5 20 30 5 1 2 4 0	60 3000

Problemsetter: Shahriar Manzoor, Special Thanks: Monirul Hasan, Md. Mahbubul Hasan

Input The input file contains at most 500 test cases. The description of each test case is given below:

First line of each test case contains an integer T (T ≤ 100) which denotes the total number of ants to consider. Each of the next T lines contains three integers which denote the value of L, W and H (1 ≤ L, W, H ≤ 50) of an ant respectively.

Input is terminated by a line containing a single zero.

Output For each set of input produce one line of output. This line contains the volume of the ant that has the highest downward acceleration. If there is a tie report the ant with the highest volume.

Sample Input Output for Sample Input

3 3 4 5 12 1 5 20 10 4 3 3 4 5 20 30 5 1 2 4 0

60 3000

Problemsetter: Shahriar Manzoor, Special Thanks: Monirul Hasan, Md. Mahbubul Hasan

B

Game of MJ

Input: Standard Input
Output: Standard Output

M and **J** are playing a game. The description of the game is very simple. They have a few displays where several digits will be shown. There are buttons under each digit of every display. If someone presses the **i^th^** button once, the value of the **i^th^** digit will increase by one. Each of these displays has one fault in common. The **0^th^** digit (least significant digit) can't be increased because its button is broken.

Each of these displays can be described using two parameters: **L** and **B**. **L** is the length of the display or the number of digits shown by the display. If **L** is **3** then the display can show **three (3)** digits side by side (we can consider it like a **3**-digit number). **B** is the base of the display. It is important in two aspects. Firstly, it limits the number of times you can press a button. Secondly, the number displayed in the display will be interpreted as a **B**-based number with **L** digits. Suppose **B** is **8** and **L** is **4**. Then the numbers shown by the display will be octal numbers and the length of the numbers will be **4**. Also you can press each of the **three** working (not broken) buttons at most **seven (7)** times. Example of such a display is shown in Figure 1.

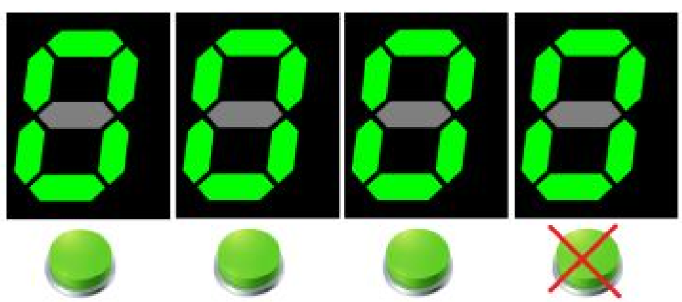


Figure 1 : A display with L = 4 and B = 8

M and **J** have **N** of these displays. Each of the display has its own **L** and **B**. Now the game **M** and **J** are playing goes like this:

1. **M** plays first, **J** plays second. After that they alternate moves.
2. In each move, a player selects a display. Then selects a digit. And presses its button. A digit can only be selected, if it hasn't reached its limit **B-1** already. A display can only be selected, if there is a digit which can be selected. However since the switch for the rightmost digit is already broken, you can not choose this button for any display.
3. After a move, if the summation of numbers (After converting it to decimal base) shown in the displays is divisible by **3**, then the player making that move loses and the other player wins.
4. If there is no move possible, then the game ends in draw.

Given description **N** displays, you need to find out the outcome of the game. **M** and **J** both plays optimally. Initially every digit of every display is set to **0** (zero).

M and J are playing a game. The description of the game is very simple. They have a few displays where several digits will be shown. There are buttons under each digit of every display. If someone presses the i

th

button once, the value of the i

th

digit will increase by one. Each of these displays has one fault in common. The 0

th

digit (least significant digit) can't be increased because its button is broken.

Each of these displays can be described using two parameters: L and B. L is the length of the display or the number of digits shown by the display. If L is 3 then the display can show three (3) digits side by side (we can consider it like a 3-digit number). B is the base of the display. It is important in two aspects. Firstly, it limits the number of times you can press a button. Secondly, the number displayed in the display will be interpreted as a B-based number with L digits. Suppose B is 8 and L is 4. Then the numbers shown by the display will be octal numbers and the length of the numbers will be 4. Also you can press each of the three working (not broken) buttons at most seven (7) times. Example of such a display is shown in Figure 1.

Figure 1 : A display with L = 4 and B = 8 M and J have N of these displays. Each of the display has its own L and B. Now the game M and J are playing goes like this:

1. M plays first, J plays second. After that they alternate moves. 2. In each move, a player selects a display. Then selects a digit. And presses its button. A digit can only be selected, if it hasn't reached its limit B-1 already. A display can only be selected, if there is a digit which can be selected. However since the switch for the rightmost digit is already broken, you can not choose this button for any display. 3. After a move, if the summation of numbers (After converting it to decimal base) shown in the

displays is divisible by 3, then the player making that move loses and the other player wins. 4. If there is no move possible, then the game ends in draw. Given description N displays, you need to find out the outcome of the game. M and J both plays optimally. Initially every digit of every display is set to 0 (zero).

B Game of MJ Input: Standard Input Output: Standard Output

Input

First line of input consists of an integer **T** (**T ≤ 100**), the number of test cases. Each test case starts with an integer **N** (**0 < N <10**), the number of displays. Next **N** line each contains two integers, **L** (**0 < L < 10^9**) and **B** (**1 < B < 10^9**) which are the parameters that describes the **i^{th}** display.

Output

For each case print one line: "`Case X: S`", where **X** is the case number. **S** is either **"M"**, **"J"** or **"Draw"** based on the outcome of the game in that case.

Sample Input	Output for Sample Input

`3`
`1`
`5 2`
`2`
`5 2`
`2 6`
`1`
`2 2`
	`Case 1: M`
	`Case 2: J`
	`Case 3: Draw`

Problemsetter: Hasnain Heickal, Special Thanks: Md. Shiplu Hawlader

Explanation

Case 1:

00000 => 00010 => 01010 => 01110 => 11110 which is 30 in decimal and divisible by 3. So J loses and M wins. However this is one possible valid game sequence. But if two players play optimally J will always lose and thus M will always win.

Case 2:

(000,00) => (000,10) Sum = 0 + 6 = 6.

(000,00) => (100,00) => (100,10) => (110,10) Sum = 6 + 6 = 12.

(000,00) => (100,00) => (100,10) => (100,20) => (100,30) => (100,40) => (100,50) => (110,50) Sum = 6 + 30 = 36

In this way, in every game sequence M is doomed to reach such a configuration where the sum will be divisible by 3. Hence, in this scenario J will win and M will lose.

Case 3:

(00) => (10) which is 2 in decimal and not divisible by 3. There is no move left. So the game ends in a draw.

Input First line of input consists of an integer T (T ≤ 100), the number of test cases. Each test case starts with an integer N (0 < N <10), the number of displays. Next N line each contains two integers, L (0 < L < 109) and B (1 < B < 10

9

) which are the parameters that describes the i

th

display.

Output For each case print one line: “Case X: S”, where X is the case number. S is either “M”, “J” or “Draw” based on the outcome of the game in that case.

Sample Input Output for Sample Input 3 1 5 2 2 5 2 2 6 1 2 2

Case 1: M Case 2: J Case 3: Draw

Problemsetter: Hasnain Heickal, Special Thanks: Md. Shiplu Hawlader

Explanation

Case 1:

00000 => 00010 => 01010 => 01110 => 11110 which is 30 in decimal and divisible by 3. So J loses and M wins. However this is one possible valid game sequence. But if two players play optimally J will always lose and thus M will always win.

Case 2:

(000,00) => (000,10) Sum = 0 + 6 = 6.

(000,00) => (100,00) => (100,10) => (110,10) Sum = 6 + 6 = 12.

(000,00) => (100,00) => (100,10) => (100,20) => (100,30) => (100,40) => (100,50) => (110,50) Sum = 6 + 30 = 36

In this way, in every game sequence M is doomed to reach such a configuration where the sum will be divisible by 3. Hence, in this scenario J will win and M will lose.

Case 3:

(00) => (10) which is 2 in decimal and not divisible by 3. There is no move left. So the game ends in a draw.

C Game of Throne

Input: Standard Input
Output: Standard Output

General Broken Arrow and Lieutenant General Shadow Coder are two great warriors of Gigaland. Every year they fought several wars together like ICPC, NCPC etc. The peoples of Gigaland were living happily until one day the king of Gigaland General Rem (RIP) died. Now both Broken Arrow and Shadow Coder wants to become the new king of Gigaland. As this is not possible for two persons to become king of a country at the same time, they leave it to the field of war. They decided to fight each other in the Great Chaos Jam (GCJ) field and the winner will become the new king.

Colonel Dragoon, friend of both Broken Arrow and Shadow Coder wants to avoid the blood shedding war and comes up with a great solution. According to his proposed solution the country will be divided into two disjoint sets of cities, called **A** and **B**. Broken Arrow will rule the cities in the set of **A** and Shadow Coder will rule the rest of the cities. But the people of Gigaland will get frustrated and angry with these divisions. To make peoples remain calm and happy Dragoon thought another idea. They will build a sub-country of Gigaland called Megaland consist of some cities and connecting roads of Gigaland so that peoples from set **A** and **B** can have some access to other cities. Though cities of set **A** and **B** are disjoint, cities might not be disjoint between Megaland and set **A** and between Megaland and set **B**. Broken Arrow and Shadow Coder both agreed with the idea of Dragoon.

The country of Gigaland consists of **N** (numbered from **1** to **N**) cities and **M** bidirectional roads each of which connects two different cities and each road has a cost to travel. Gigaland was built such a way that every city is connected to each other with sequence of one or more roads. Set **A** will consists of first **K(2 ≤ K ≤ min(N,50))** cities, **1, 2, … , K** numbered cities of Gigaland and rest **N – K** cities will belong to the set **B**.

Though the warriors have agreed to build the new sub-country Megaland, they have some conditions. Each city of set **A** should be incident to **odd** number of roads of Megaland and each city of set **B** should be incident to **even** number (possibly zero) of roads of Megaland. The cost of road system in Megaland should be minimum, which is sum of all the road cost of Megaland should be minimized. The cities of Megaland might or might not be connected to each other.

In terms of graph theory, you are given a weighted graph **G = (N, M)**, two set of nodes **A = {1, 2, …, K}** and **B = {K+1, K+2, …, N}**. You have to find the minimum cost sub-graph **S (subset of edges)**, where each node in **A** should have odd degree in **S** and each node in **B** should have even degree in **S**. Here cost of **S**, is sum of all the edge cost in **S**.

Given **N, M, K** and description of **M** roads of Gigaland, find the minimum possible cost of sub-country Megaland.

General Broken Arrow and Lieutenant General Shadow Coder are two great warriors of Gigaland. Every year they fought several wars together like ICPC, NCPC etc. The peoples of Gigaland were living happily until one day the king of Gigaland General Rem (RIP) died. Now both Broken Arrow and Shadow Coder wants to become the new king of Gigaland. As this is not possible for two persons to become king of a country at the same time, they leave it to the field of war. They decided to fight each other in the Great Chaos Jam (GCJ) field and the winner will become the new king.

Colonel Dragoon, friend of both Broken Arrow and Shadow Coder wants to avoid the blood shedding war and comes up with a great solution. According to his proposed solution the country will be divided into two disjoint sets of cities, called A and B. Broken Arrow will rule the cities in the set of A and Shadow Coder will rule the rest of the cities. But the people of Gigaland will get frustrated and angry with these divisions. To make peoples remain calm and happy Dragoon thought another idea. They will build a sub-country of Gigaland called Megaland consist of some cities and connecting roads of Gigaland so that peoples from set A and B can have some access to other cities. Though cities of set A and B are disjoint, cities might not be disjoint between Megaland and set A and between Megaland and set B. Broken Arrow and Shadow Coder both agreed with the idea of Dragoon.

The country of Gigaland consists of N (numbered from 1 to N) cities and M bidirectional roads each of which connects two different cities and each road has a cost to travel. Gigaland was built such a way that every city is connected to each other with sequence of one or more roads. Set A will consists of first K($2 \le K \le \min(N,50)$) cities, 1, 2, ... , K numbered cities of Gigaland and rest N – K cities will belong to the set B.

Though the warriors have agreed to build the new sub-country Megaland, they have some conditions. Each city of set A should be incident to odd number of roads of Megaland and each city of set B should be incident to even number (possibly zero) of roads of Megaland. The cost of road system in Megaland should be minimum, which is sum of all the road cost of Megaland should be minimized. The cities of Megaland might or might not be connected to each other.

In terms of graph theory, you are given a weighted graph G = (N, M), two set of nodes A = {1, 2, ..., K} and B = {K+1, K+2, ..., N}. You have to find the minimum cost sub-graph S (subset of edges), where each node in A should have odd degree in S and each node in B should have even degree in S. Here cost of S, is sum of all the edge cost in S.

Given N, M, K and description of M roads of Gigaland, find the minimum possible cost of sub-country Megaland.

C Game of Throne

Input: Standard Input Output: Standard Output

Input

Input starts with a positive integer, **T (T ≤ 50)** denoting the number of test cases. Each case starts with three integers, **N (2 ≤ N ≤ 100), M (N-1 ≤ M ≤ N*(N-1)/2)** and **K (2 ≤ K ≤ min(N,50))**. Each of the next **M** lines contains three integers **u**, **v** and **c (1 ≤ u,v ≤ N, u ≠ v, 0 < c < 10000)** meaning that there is a bi-directional road between city **u** and **v** of cost **c**. No two roads will be same.

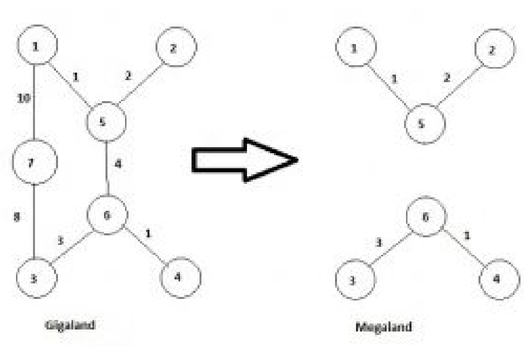
Output

For each test case, print the test case number (starting from **1**) and the minimum possible cost of Megaland if it is possible to build the sub-country according to the conditions described above, otherwise print "Impossible".

Sample Input	Output for Sample Input
2 7 7 4 1 7 10 1 5 1 2 5 2 3 6 3 4 6 1 5 6 4 3 7 8 4 4 3 1 2 1 1 3 1 2 4 1 3 4 1	Case 1: 7 Case 2: Impossible

Explanation

For test case 1,



In Megaland degree of each node in set A = {1,2,3,4} is 1 and degree nodes in set B = {5, 6, 7} are 2, 2 and 0 respectively.

Problemsetter: Md. Shiplu Hawlader, Special Thanks: Rujia Liu, Md. Mahbubul Hasan

Input Input starts with a positive integer, T (T ≤ 50) denoting the number of test cases. Each case starts with three integers, N (2 ≤ N ≤ 100), M (N-1 ≤ M ≤ N*(N-1)/2) and K (2 ≤ K ≤ min(N,50)). Each of the next M lines contains three integers u, v and c (1 ≤ u,v ≤ N, u ≠ v, 0 < c < 10000) meaning that there is a bi- directional road between city u and v of cost c. No two roads will be same.

Output For each test case, print the test case number (starting from 1) and the minimum possible cost of Megaland if it is possible to build the sub-country according to the conditions described above, otherwise print “Impossible”.

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

Case 1: 7 Case 2: Impossible

Explanation For test case 1,

In Megaland degree of each node in set A = {1,2,3,4} is 1 and degree nodes in set B = {5, 6, 7} are 2, 2 and 0 respectively.

Problemsetter: Md. Shiplu Hawlader, Special Thanks: Rujia Liu, Md. Mahbubul Hasan

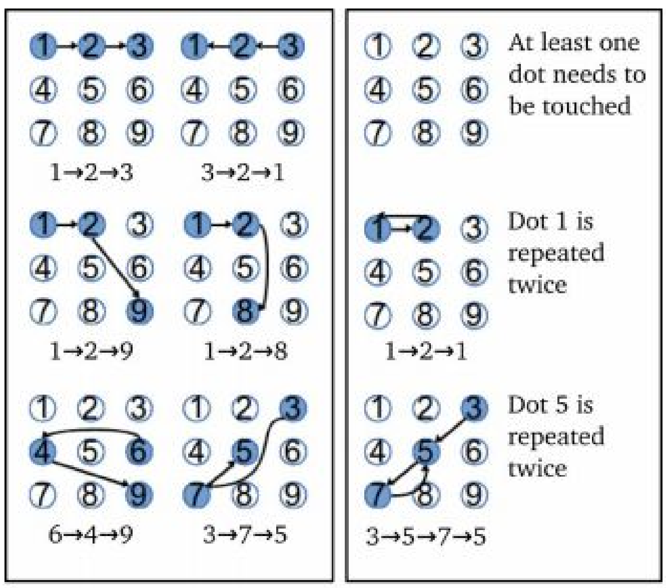
D — Pattern Locker

Input: Standard Input
Output: Standard Output

Mr. Anderson is in a relationship with a very suspicious and jealous girlfriend. She is always checking Anderson's phone logs and texts to find out he is up to. Feeling that his private space is getting violated, he decided to put a pattern locker on his phone.

This pattern locker comes with a **9** dots arranged on a **3x3** square grid by default. One has to drag through several dots to record a pattern. Then to unlock the phone, one needs to replicate the same pattern recorded before. If we assume that each dot is assigned a unique number then a pattern is nothing more than a sequence of digits. The pattern locker requires that no digit appears in the sequence more than once.

Here is an example of such a pattern locker and some valid and invalid recorded pattern:



Valid patterns touching 3 dots on a 3x3 grid

Invalid patterns on a 3x3 grid

Even after recording a hard to crack pattern, Mr. Anderson doesn't feel quite comfortable. He is worried that his girlfriend might try all possible sequences to break his pattern. He wants to know how many different pattern sequences are possible for a given grid size, minimum and maximum numbers of dots recorded in patterns.

You have to help Mr. Anderson in counting the number of possible such sequences.

Mr. Anderson is in a relationship with a very suspicious and jealous girlfriend. She is always checking Anderson's phone logs and texts to find out he is up to. Feeling that his private space is getting violated, he decided to put a pattern locker on his phone.

This pattern locker comes with a 9 dots arranged on a 3x3 square grid by default. One has to drag through several dots to record a pattern. Then to unlock the phone, one needs to replicate the same pattern recorded before. If we assume that each dot is assigned a unique number then a pattern is nothing more than a sequence of digits. The pattern locker requires that no digit appears in the sequence more than once.

Here is an example of such a pattern locker and some valid and invalid recorded pattern:

Even after recording a hard to crack pattern, Mr. Anderson doesn't feel quite comfortable. He is worried that his girlfriend might try all possible sequences to break his pattern. He wants to know how many different pattern sequences are possible for a given grid size, minimum and maximum numbers of dots recorded in patterns.

You have to help Mr. Anderson in counting the number of possible such sequences.

D Pattern Locker

Input: Standard Input Output: Standard Output

Input

The first line of the input will give the number of test cases, **T (1 ≤ T ≤ 10000)**. Then **T** test cases follow in separate lines. Each test case consists of three numbers **L**, **M**, **N** separated by a single space in between two numbers. The first number **L (1 < L ≤ 100)** denotes the number of rows and columns in the grid. The second number **M (1 ≤ M ≤ L*L)** denotes minimum number of dots to be included in a pattern and the third number **N (M ≤ N ≤ L*L)** denotes the maximum number of dots to be included in the pattern.

Output

For each test case, you need to print the test case number **X** in the format "Case X: ". This will be followed by the count of possible sequences for the given grid size, minimum and maximum number of dots in a sequence. Since the count can be pretty big, you need to print the value of the count modulo **10000000000007** (**1** followed by **12** zeros followed by **7**).

Sample Input	Output for Sample Input
2 3 4 9 3 1 9	Case 1: 985824 Case 2: 986409

Problemsetter: Monirul Hasan, Special Thanks: Shahriar Manzoor, Md. Mahbubul Hasan

Input The first line of the input will give the number of test cases, T (1 ≤ T ≤ 10000). Then T test cases follow in separate lines. Each test case consists of three numbers L, M, N separated by a single space in between two numbers. The first number L (1 < L ≤ 100) denotes the number of rows and columns in the grid. The second number M (1 ≤ M ≤ L*L) denotes minimum number of dots to be included in a pattern and the third number N (M ≤ N ≤ L*L) denotes the maximum number of dots to be included in the pattern.

Output For each test case, you need to print the test case number X in the format “Case X: ”. This will be followed by the count of possible sequences for the given grid size, minimum and maximum number of dots in a sequence. Since the count can be pretty big, you need to print the value of the count modulo 10000000000007 (1 followed by 12 zeros followed by 7).

Sample Input Output for Sample Input 2 3 4 9 3 1 9

Case 1: 985824 Case 2: 986409

Problemsetter: Monirul Hasan, Special Thanks: Shahriar Manzoor, Md. Mahbubul Hasan