A. Solving != Coding

Score: 1

CPU: 2s
Memory: 1024MB

Many people think that programming contest is waste of time; they think coding is same as solving. Let’s give them a hard time. Let’s make them understand solving is not coding, coding is much much easier than solving and solving is not for dumb coders.

Anyway, I think you are quite familiar with minions. If not you should watch this, funny no? May be annoying to some people but who cares if it helps you to joy the Mount Everest! Here are the lyrics:

```
BA BA BA BABANANA
BA BA BA BABANANA

BANANA NA AHH, POTATO NA AH AH BANANA AH AH

TO GA LI NO PO TAH TO NI GAH NI BAH LO BAH NI KAH NO JI GAH BA BA BA BABANANA.

YO PLANO HU LA PA NO NO TU MA BANANA LIKE A NUPI TALAMOO

BANANA BA BA BABANANA BA BA BA BABANANA

POTATO HO HOOOOOO

TO GA LI NO PO TAH TO NI GAH NI BAH LO BAH NI KAH NO JI

GAH BA BA BA BABANANAAAAAAAAA

DHISYAAAAAA
```

Your task is to print number of letter “A” in these lyrics. Those who are coders go and code a sophisticated solution. Those who are solvers show why you do programming contest!

Input

No input

Output

Just output the number of letter “A” in the lyrics.

Sample

Input	**Output**
	95

(This output may give wrong answer; this is just to show how the output may look like.)

Hint:

Code snippet in C/C++ to print the above Sample Output

```
#include<stdio.h>
int main()
{
    printf(“95\n”);
    return 0;
}
```

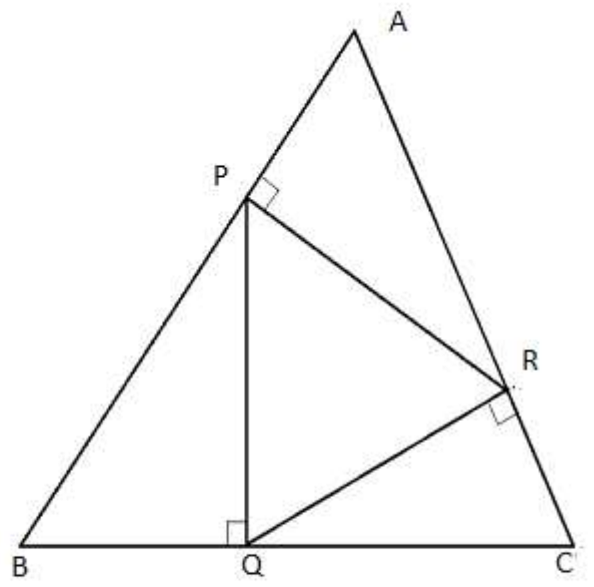
You just need to replace **95** with the correct answer.

B. Triangle in Triangle

Score: 1

CPU: 2s
Memory: 1024MB

You are given the lengths of three sides of an acute triangle **ABC** as shown below. Assume a point **P** on the side **AB** such that, if **Q** is the projection of **P** onto **BC**, **R** is the projection of **Q** onto **CA**, **P** becomes the projection of **R** onto **AB**. Find the length **PB**.



Input

There will be **T** test cases, at most **10,000**. For each case, your program will read three integer values: the lengths of the sides **BC**, **CA** and **AB** respectively. Every integer in the input file is positive and no larger than **10,000**. Triangle **ABC** is guaranteed to be acute.

Output

For each test case, print the case number along with the length of **PB**. You can print any number of digits after decimal point. Floating point rounding error less than **1e-5** will be ignored by the judge.

Sample

Input

```
5
10 10 10
11 10 10
5 6 7
6 7 5
7 5 6
```

Output

```
Case 1: 6.66666667
Case 2: 7.53894081
Case 3: 3.18181818
Case 4: 3.27272727
Case 5: 5.34545455
```

Note: An acute triangle is a triangle where all three internal angles are acute i.e. less than **90** degrees.

C. Movie Show

Score: 1

CPU: 2s
Memory: 1024MB

You are given a list of movie titles to choose from to do a movie show at your school for a fund raiser. The movies you can pick are classified into four different genres and they have a set run time, as well. You were told that you have to finish the movie show within a specific amount of time. Now all you need to do is to pick four movies, one from each genre so that you can run the show within the stipulated time.

Of course, there can be many valid selection possibilities. You'll have to ensure that you pick a selection that maximizes the total run time of the show. Even then there could be many different valid selections that equal the maximum possible run time. Any of those would do.

Input

The input description for the problem will start with **T** (1 < T <= 100) – the number of test cases. Then **T** test cases will follow. Each test case will start with the value of total duration of the show in minutes, **D** (0 < D < 2000). The next line will give you **N** (3 < N < 41), the number of movies to choose from. Each of the next **N** lines will print the information for a movie. Each movie description starts with the title of the movie (a sequence of letters: only **A~Z, a~z, 0~9** and **underscores** are allowed), followed by its genre (any one of **Action**, **Comedy**, **Horror** or **Animation**), followed by its run time in minutes, **R** (0 < R < 300). The title of each movie will contain at most **40** letters. You can safely assume that each line of a movie description describes a different movie (even if the same movie name was mentioned before) and there will be at least one Action, one Comedy, one Horror and one Animation movie in the list.

Output

Output for each test case will start with the movie show label (starting with **1**, and formatted as shown in sample output.) The label will be followed by the shows total run time without exceeding the maximum allowed time. When such a selection is not possible, you need to print "Movie show canceled!" without the quotes.

Sample

Input	**Output**
3 610	Case 1: 610 Case 2: Movie show canceled!

Input

```
10
Most_Welcome Comedy 240
Most_Welcome_2 Comedy 180
Mission_Impossible Action 100
Frozen Animation 120
Ice_Age Animation 100
Mama Horror 200
I_Dont_Care Comedy 200
Jhole_Mon_Gacher_Dale Action 110
Dhil_Mari_Tor_Tiner_Chale Action 210
Amti_Ami_Khabo_Pere Animation 240
560
10
Most_Welcome Comedy 240
Most_Welcome_2 Comedy 180
Mission_Impossible Action 100
Frozen Animation 120
Ice_Age Animation 100
Mama Horror 200
I_Dont_Care Comedy 200
Jhole_Mon_Gacher_Dale Action 110
Dhil_Mari_Tor_Tiner_Chale Action 210
Amti_Ami_Khabo_Pere Animation 240
580
10
Most_Welcome Comedy 240
Most_Welcome_2 Comedy 180
Mission_Impossible Action 100
Frozen Animation 120
Ice_Age Animation 100
Mama Horror 200
I_Dont_Care Comedy 200
Jhole_Mon_Gacher_Dale Action 110
Dhil_Mari_Tor_Tiner_Chale Action 210
Amti_Ami_Khabo_Pere Animation 240
```

Output

```
Case 3: 580
```

D. National Treasure

Score: 1

CPU: 3s
Memory: 1024MB

Benjamin Franklin Gates (Ben) is an American historian, cryptologist and the youngest descendent of long line of treasure hunters. This time Ben is on the way to find a secret treasure and he is driven by a written clue given by his grandfather "The secret lies with Charlotte". According to family history, the clue could lead to the fabled "National Treasure" fought over since Ancient Egypt and hidden by the Founding Fathers and Freemasons during the American Revolutionary War.

At last Ben found the area where the treasure has been kept. The area is an axis parallel rectangular grid with east and west side closed and north and south side open. Some grid cells contain trees and only one grid cell have the treasure. Ben knows the position of the treasure in the grid but he can't just walk into the grid and collect the treasure because surface of the grid is full of garbage and Ben can get hurt if he walks into the grid.

There is a good and a bad news for Ben at this moment. The good news is Ben has brought his skateboard and he is going to collect the treasure from the grid using his skateboard. The bad news is he forgot how to control the skateboard. He can only start skating from a position in the northern row of the grid and go in a straight line from north to south direction but cannot stop it. If there is a tree in the line of his skating he will hit the tree and stop in the cell containing the tree. Otherwise he will get out of the grid. Ben can never pass through a tree cell and from a tree cell he can't start his skating again. Ben can jump from a cell (tree or non-tree) to its two neighbor cells (east cell and west cell). It does not matter if the cell he just jumped into contains a tree or not. By one or more jumps he should go to a non-tree cell (if there is any) start his skating again to the south. Please note he can't jump while skating. Also he will not hit a tree when jumping to a tree cell. To capture the treasure Ben has to go through the treasure cell or its west neighbor cell or east neighbor cell. Note that, the treasure is so small that Ben will not hit the treasure and stop in the treasure cell when skating but he can collect it on the way.

Ben will start from a cell of the north row of the grid and collect the treasure and get out from the south side with minimum number of hit with trees. He can't get out of the grid from west and east side of the grid.

Given the information of the grid, find the minimum number of hits Ben will get with tree to collect the treasure and get out of the grid.

Input

First line contains an integer **T** (T <= 100), number of test case. First line of each test case contains three integers **N**, **M** and **K** (1 <= N, M <= 1,000,000,000 and 0 <= K <= 1,000), number of rows from north to south, number of columns from west to east and number of tree cell in the grid respectively. Next line will contain two integers **Rx** and **Ry** (1 <= Rx <= N and 1 <= Ry <= M) denoting the position of the treasure, where **Rx** is the row number and **Ry** is the column number of the treasure. Next **K** lines will contain two space separated integers $\mathbf{X_i}$ and $\mathbf{Y_i}$ (1 <= X_i <= N and 1 <= Y_i <= M), position of the i-th tree, where $\mathbf{X_i}$ is the row number and $\mathbf{Y_i}$ is the column number. The North West corner of the grid is recognized as **(1, 1)** and south east corner as **(N, M)**, that means grid rows goes from north to south and column goes from west to east.

The treasure cell will not contain a tree.

Output

For each test case print the test case number and “Impossible” without quotes if it is not possible to collect the treasure and get out of the grid. Otherwise print the minimum number of hits Ben has to get with trees.

Sample

Input	**Output**
4	Case 1: 1
5 5 4	Case 2: 0
4 3	Case 3: Impossible
2 2	Case 4: 0
2 3	
3 4	
5 3	
5 5 4	
4 3	
2 2	
2 3	
3 3	
5 3	
5 5 5	

Input	Output
4 3	
2 2	
2 3	
3 2	
3 3	
3 4	
5 5 5	
1 3	
1 1	
1 4	
2 2	
2 3	
4 4	

E. Date Puzzle

Score: 1

CPU: 2s
Memory: 1024MB

Weekends at abroad are not same as in Bangladesh. Here weekends are for many different activities. Some day we spend at museum, someday at exhibition, someday at lake and so on. There are many websites where we can find out the date range of some events. However understanding the date is sometimes difficult. For example: say an event date range is given as: from **1/2** to **4/3**. What will you understand? From second January or first February? Similarly till when? Third April or fourth march? Sometimes even though one of them is ambiguous by its own but looking at both of them it becomes clear. For example: from **1/2** to **1/20**. Here the first one is ambiguous by itself. However, the second one is of course 20th January and since we expect both of them to be in the same format the first one is second January not first February.

So your task is: given a date range, you have to tell me if you can figure out the exact dates. For your convenience, you may consider the dates are from 2014 only. Please note sometimes because of typo the dates may be invalid for example a date like **13/14** cannot be interpreted any way, also date range from **3/4** to **1/2** does not make sense. So is the range: **1/1** to **1/1** because we expect the first date to be strictly before the second date. But say if the range is from **5/4** to **10/1** then we will know that it is from fourth of May to first October because we expect valid date range.

So to sum up, if there is multiple possible meaning of the date range it is ambiguous, if the date range is not possible it is mistake and otherwise it is meaningful.

Input

First line contains a positive integer **T** which denotes number of test cases (T <= 540,000). Hence **T** lines follow. Each line will contain four positive integers: $\mathbf{d_1}$, $\mathbf{m_1}$, $\mathbf{d_2}$, $\mathbf{m_2}$ and none of them will exceed **40**. $\mathbf{(d_1, m_1)}$ refers to the first date and $\mathbf{(d_2, m_2)}$ refers to the second date. However, as described above $\mathbf{d_i}$ does not necessarily mean date, and $\mathbf{m_i}$ does not necessarily mean month. But if $\mathbf{d_1}$ is date and $\mathbf{m_1}$ is month, then so are $\mathbf{d_2}$ and $\mathbf{m_2}$. And similarly if $\mathbf{d_1}$ is month and $\mathbf{m_1}$ is date, so are $\mathbf{d_2}$ and $\mathbf{m_2}$.

Output

For each test case, first output case number and then output “Okay got it!” or “I am sure there is some kinda mistake!” or “Oh no it is ambiguous!” depending on the situation. Follow the sample for clarity.

Sample

Input	Output

Input

```
5
1 2 1 20
1 2 4 3
3 4 1 2
13 14 1 2
1 1 1 1
```

Output

```
Case 1: Okay got it!
Case 2: Oh no it is ambiguous!
Case 3: I am sure there is some kinda mistake!
Case 4: I am sure there is some kinda mistake!
Case 5: I am sure there is some kinda mistake!
```

Month = 1 refers to January and Month = 12 refers to December (and so are in between).

Here is the number of days in the months from 2014: January (31), February (28), March (31), April (30), May (31), June (30), July (31), August (31), September (30), October (31), November (30), December (31).

F. Coin Change

Score: 1

CPU: 2s
Memory: 1024MB

Given **N** coins, find out how many permutations of the coins have a prefix (possibly empty) whose sum is equal to **K**. All the coins are considered different even if they have the same value. For example, given coins 1, 2, 3, 3, 6 (N = 5) and K = 5 the permutations we are looking for are given below. For better understanding, the first **3** is denoted by **3a** and second three is denoted by **3b**.

2 3b 1 3a 6	**2 3a** 1 3b 6	**3a 2** 1 3b 6	**3b 2** 1 3a 6
2 3b 1 6 3a	**2 3a** 1 6 3b	**3a 2** 1 6 3b	**3b 2** 1 6 3a
2 3b 3a 1 6	**2 3a** 3b 1 6	**3a 2** 3b 1 6	**3b 2** 3a 1 6
2 3b 3a 6 1	**2 3a** 3b 6 1	**3a 2** 3b 6 1	**3b 2** 3a 6 1
2 3b 6 1 3a	**2 3a** 6 1 3b	**3a 2** 6 1 3b	**3b 2** 6 1 3a
2 3b 6 3a 1	**2 3a** 6 3b 1	**3a 2** 6 3b 1	**3b 2** 6 3a 1

Here, two coins with value **3** are considered different. Prefixes with sum = K are underlined. So in this case the result will be **24**.

Input

First line, **T** (T <= 100), number of test cases. Each case will start with **N** (0 <= N <= 32) and **K** (0 <= K <= 10^{12}). Next line will contain N integers, value of the coins which will be positive and less than or equal to **10^9**.

Output

For each case, output one line: “Case C: A”, where **C** is the case number and **A** is the answer for the case modulo **1,000,000,007**.

Sample

Input

```
2
5 5
1 2 3 3 6
5 6
1 2 3 3 6
```

Output

```
Case 1: 24
Case 2: 60
```

G. The Tri-wizard Tournament

Score: 1

CPU: 6s
Memory: 1024MB

Harry Potter is going to face a tough task in the Triwizard tournament. He has to go through a magical chessboard on a knight. The knight move is similar to the chess played by muggles – from the current square, it can move to a square which is horizontally 2 squares away and vertically 1 square away, or vertically 2 squares away and horizontally 1 square away. Obviously, it cannot go out of the board. There are some *dangerous squares* where it cannot go. Moreover, Harry needs to collect some magical weapons to complete the task. The position of those weapons will be known. He has to use all of them when he will approach the next task, so he has to collect all of the weapons.

There are also some dragons flying over the board which will constantly try to follow Harry. After reading a lot about these dragons, Hermione discovers an interesting fact. If he *does not change the direction in consecutive moves*, then it will be too easy for those dragons to kill him. Here, direction means the signed change in vertical and horizontal axis. If both of the signed value changes in vertical and horizontal direction remain same (in consecutive moves), it will be the end of his journey. For example, if he is currently in **(2, 3)** and came here from **(1, 1)**, then he cannot go to **(3, 5)**. He may go to **(4, 4)**, **(0, 2)**, **(3, 1)**, **(4, 2)**, **(0, 4)**, **(1, 5)** or even go back to **(1, 1)**. The amount of time required to complete one move also depends on the direction of the move. If the sign of change in both directions remain same, it takes no time (i.e. **0** seconds) to complete the move, otherwise it takes **1** second. For example, from **(4, 4)**, it would take **0** second to go to **(6, 5)**, **(5, 6)**, **(2, 3)** or **(3, 2)**, but it would take **1** second to go to **(3, 6)**, **(6, 3)**, **(2, 5)** or **(5, 2)**.

Harry needs to complete this task as soon as possible by collecting all the weapons. Hermione could easily write a program to calculate the minimum time, but she is very busy with her research to understand the proper usage of all these new weapons. So she wants your help.

Input

The first line of input contains a single integer **T** (1 <= T <= 120), which denotes the number of test cases to follow. For each test case, the first line contains **3** integers: **m** (1 <= m <= 100), **n** (1 <= n <= 100) and **d** (0 <= d <= m * n). Here **m** denotes the number of rows of the board, **n** denotes the number of columns of the board and **d** denotes the number of dangerous squares. Then **d** lines follow, where each line contains two integers: **r** and **c**, meaning **(r, c)** is the position of a dangerous square. Then there will be one line containing one integer **w** (0 <= w <= 5), which denotes the number of weapons. Then **w** lines follow, where each line contains two integers: **r** and **c**, meaning **(r, c)** is the position of a weapon. All row and columns are **0**-indexed. No square will contain multiple weapons.

Output

For each case, in a separate line, print the case number and the minimum amount of time required to start the journey from **(0, 0)** and end at **(m – 1, n – 1)** after collecting all the weapons. If it is not possible to complete the task, print "Impossible" (without quotes). Follow Sample Input and Output for details.

Sample

Input

```
3
3 3 0
2
0 2
2 1
3 3 1
1 0
2
0 2
2 1
3 3 1
1 0
1
1 1
```

Output

```
Case 1: 2
Case 2: 4
Case 3: Impossible
```

H. Fallen Chomp

Score: 1

CPU: 2s
Memory: 1024MB

In a cloudy autumn afternoon, Mr. Fallen felt to write some poems. As you might already have known he is very poor at poetry. So after quite some trying he changed his mind and tried to learn a new game- Chomp.

The game chomp is a two player game, played in a 2D rectangular board. The board has **M** rows and each of the rows has **N** columns initially, so **N * M**cells in total. You can consider that the leftmost bottom cell of the board is at co-ordinate **(1, 1)** and the rightmost top cell is at co-ordinate **(N, M)**. In a turn, a player should select a cell of the board and all the cells with both **X** and **Y** co-ordinates greater than or equal to the selected cell gets removed from the board. The turn of the player alters and the one with the last move loses.

After trying the game for a while Mr. Fallen did understand that this game is too hard for him! Instead he thought if he could play with arbitrary valid moves, how many different boards could he see throughout the game including the empty one.

He tried and tried and kept failing and finally he came up with the smart idea to use brilliant mind like you to help him solving this problem. Help him to find total number of different boards possible to make from the given board with valid moves. Two boards will be considered different if a cell exists in one but not in the other.

Input

The first line of the input contains an integer **T** (T <= 100,000) denoting the number of test cases. Each of the following **T** lines has two space separated integers **M** and **N** (1 <= M, N <= 100,000).

Output

For each input, print the output in the format, “Case C: X” (quote for clarity), where **X** is the number of different boards possible including the empty board modulo **1,000,000,007**.

For exact output format please check sample input/output section.

Sample

Input	Output
2	Case 1: 2
1 1	Case 2: 3
1 2	