2011阿里巴巴程序设计公开赛

-----NotOnlySuccess 专场

1001 Coin Game

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

After hh has learned how to play Nim game, he begins to try another coin game which seems much easier.

The game goes like this:
Two players start the game with a circle of n coins.
They take coins from the circle in turn and every time they could take 1~K continuous coins.
(imagining that ten coins numbered from 1 to 10 and K equal to 3, since 1 and 10 are continuous, you could take away the continuous 10 , 1 , 2 , but if 2 was taken away, you couldn't take 1, 3, 4, because 1 and 3 aren't continuous)
The player who takes the last coin wins the game.
Suppose that those two players always take the best moves and never make mistakes.
Your job is to find out who will definitely win the game.

Input

The first line is a number T(1<=T<=100), represents the number of case. The next T blocks follow each indicates a case.
Each case contains two integers N(3<=N<=10^9,1<=K<=10).

Output

For each case, output the number of case and the winner "first" or "second".(as shown in the sample output)

Sample Input

```
2
3 1
3 2
```

Sample Output

```
Case 1: first
Case 2: second
```

1002 Fruit Ninja

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Fruit Ninja is a popular classic game. During the game, fruits will up to the air, and your aim is cut as more fruits as possible with a line.

Even if the line touch a point of a fruit, the fruit also be cut.

Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.
The first line of each case contains one integer N (1<=N<=10)
Then N lines follow, each line contains a integer K(3<=K<=10), represent the number points of the fruit, then K*2 integers follow, each two integers represent one point of the fruit.(with anticlockwise order)
I promise all fruits are convex polygon, and any two fruit have no common point.

Output

For each case, output the number of case and the maximum fruits you could cut with a line.(as shown in the sample output)

Sample Input

```
2
3
3 0 0 1 0 1 1
3 1 2 2 1 2 2
3 3 1 3 0 4 0
3
4 0 0 1 0 1 1 0 1
4 2 0 3 0 3 1 2 1
```

```
4 0 99 1 99 1 100 0 100
```

Sample Output

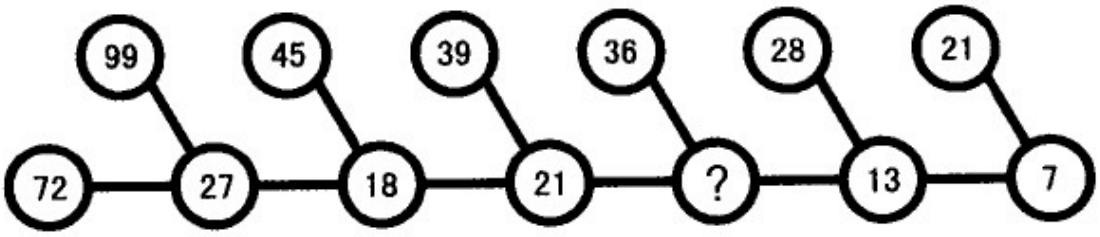
```
Case 1: 3
Case 2: 2
```

1003 I'll play a trick on you

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Please look the picture carefully. Then I'll give you two integers and your task is output the third one.
Please never doubt the picture.



Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.
Each case contains two integers A,B (1<=B<=A<=10^{100})

Output

For each case, output the number of case and the third integer.(as shown in the sample output)

Sample Input

```
3
99 72
45 27
39 18
```

Sample Output

```
Case 1: 27
Case 2: 18
Case 3: 21
```

Hint

If you have any idea to work out the ? and explain why but couldn't get Accepted , please email me (notonlysuccess@gmail.com), the first person will get 100RMB from me.

1004 Level up

Time Limit: 10000/3000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Level up is the task of all online games. It's very boooooooooring. There is only level up in those games, except level up.
In a online game, there are N heroes numbered id from 1 to N, each begins with level 1 and 0 Experience. They need to kill monsters to get Exp and level up.

There are many waves of monsters, each wave, the heroes with id from l_i to r_i will come to kill monsters and those hero with level k will get e_i*k Exp. If one hero's Exp reach $Need_k$ then the hero level up to level k immediately.
After some waves, I will query the maximum Exp from l_i to r_i.
Now giving the information of each wave and $Need_k$, please tell me the answer of my query.

Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.
The first line of each case contains three integers N(1<=N<=10000), K(2<=K<=10) and QW(1<=QW<=10000)each represent hero number, the MAX level and querys/waves number.
Then a line with K -1 integers, $Need_2$, $Need_3$...$Need_k$.(1 <= $Need_2$ < $Need_3$ < ... < $Need_k$ <= 10000).
Then QW lines follow, each line start with 'W' contains three integers l_i r_i e_i (1<=l_i<=r_i<=N , 1<=e_i<=10000); each line start with 'Q' contains two integers l_i r_i (1<=l_i<=r_i<=N).

Output

For each case, output the number of case in first line.(as shown in the sample output)
For each query, output the maximum Exp from l_i to r_i.
Output a black line after each case.

Sample Input

```
2
3 3 5
1 2
W 1 1 1
W 1 2 1
Q 1 3
W 1 3 1
Q 1 3

5 5 8
2 10 15 16
W 5 5 9
W 3 4 5
W 1 1 2
W 2 3 2
Q 3 5
W 1 3 8
Q 1 2
Q 3 5
```

Sample Output

```
Case 1:
3
6

Case 2:
9
18
25
```

Hint

```
Case 1:
At first ,the information of each hero is 0(1),0(1),0(1) [Exp(level)]
After first wave, 1(2),0(1),0(1);
After second wave, 3(3),1(2),0(1);
After third wave, 6(3),3(3),1(1);
Case 2:
The information of each hero finally:
18(5) 18(5) 25(5) 5(2) 9(2)
```

1005 March

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

During a game of Civilization V, much of your time will be spent moving units around the world. You'll be marching your army units off to discover stuff or to fight with your neighbors.

The world is comprised of hexagons. Generally, units move from hexagonto hexagon, paying the "Movement Cost" required to enter the new hexagon.

Movement Points:

All army units have a certain number of "Movement Points"(MPs) that they can expend on movement in every turn. Once they've expended those MPs, they can't move any more until the next turn.

Expending MPs:

Units expend MPs to enter tiles. The terrain of the tile determines the MP cost of the move. It doesn't cost anything to leave your current tile; the MP cost is determined by the tile you're entering. A unit can always move one tile if it has any MPs left.It doesn't matter how expensive the tile is; as long as the unit has some MPs left, it can enter.

Terrain of the tile:

Open terrain like Grassland and Plains cost 1 MP to enter, while Forest and Jungle costs 2.

Zones of Control:

Enemy units exert a "Zone of Control"(ZOC) over the tiles around them. When a unit moves between two tiles within an enemy's ZOC it expends all of MPs.(you can't move on the tile which contain an enemy)

Road:

As long as the unit moves from one tile containing a road into another tile containing a road,the unit will expend just 0.25 MPs no matter what terrain of the tile.(unless within an enemy's ZOC)

Rivers:

Rivers are between two tiles.
As long as the unit moves cross a rivers from a tile to another tile it expends all of MPs(unless these tiles contain roads).

Giving the information of the World, please tell me the minimum turns to cost for reaching the destination.

Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.
The first line of each case contains three integers N, M(2<=N,M<=M), indicating the size of World, and MPs(1<=MPs<=10).
Then N lines follow, each line contains M integers.
Each module number tells the information of the tile and is the sum of up to ten integers:
1: Grassland and Plains
2: Forest and Jungle
4: Road
8: Enemy
16: A river on northeast of the tile
32: A river on east of the tile
64: A river on southeast of the tile
128: A river on southwest of the tile
256: A river on west of the tile
512: A river on northwest of the tile
(each tile must contain either 1 or 2)
Then a line with two coordinates x1,y1,x2,y2(0<=x1,x2<n,0<=y1,y2<m) indicating the source and destination. There is no enemy in source and destination and source is different from destination.
The picture below shows the coordinate of the World.

Output

For each case, output the number of case and the minimum turns to cost for reaching the destination.If can't reach the destination, just output -1.(as shown in the sample output)

Sample Input

```
3
2 2 10
97 257
513 1
0 0 1 1

2 2 1
101 262
513 2
0 0 1 1
```

```
3 3 2
5 5 5
10 10 5
1 1 5
0 0 2 2
```

Sample Output

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

Hint

```
Case 1: (0,0)->(0,1)->(1,1)
First turn, you expends all of MPs to cross the river. So you need
waiting second turn to reach (1,1)
Case 2: (0,0)->(0,1)->(1,1)
There are roads on these tiles, so you cross the river expends a
quarter of MPs, then expends the remain MPs to reach (1,1).
Case 3: (0,0)->(0,1)->(0,2)->(1,2)->(2,2)
Even if there are roads on these tiles, but there are all in ZOC, so
you will expends all of MPs each move.
```

1006 SanguoSHA

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

SanguoSHA is a very popular game. The rules of sanguosha are too complex for me to describe it in English, so just let me simplify it a lot. ^ ^.

In this custom game we have several rules shown below, and you should read carefully:

Part I--Identifies:

1): There are at most 8 players while at least 4;
2): There are 4 roles: Lord, Minister, Rebel, and Traitor.
3): The goal of the Lord and Ministers is to kill all of the Rebels and the Traitor.
4): The goal of the Rebel is to kill the Lord.
5): The goal of the Traitor is to make himself alive until the game over.
6): There is only 1 Lord.
7): There is only 1 Traitor.
8): There will be at least 1 Rebel.
9): There will be at least 1 Minister.
10): Every player know the remain total of Rebel and Minister.

Part II--How to determine the game over and who wins:

1): If the Lord is alive, and there is neither alive Rebel nor Traitor, then the game over, and the Lord and Ministers win.
2): When the Lord is killed by someone, the game over, if there is no one but the Traitor alive, then the Traitor wins; otherwise, the Rebels win.

Part III--How to play in this custom game:

1): At first, the roles of all players except for the Lord are hidden, they only know their own roles. That means players have to conjecture others' role by analysing their behavior.
2): In each round, all alive players play one by one. The player1 plays first, then player2...
3): In each player's turn, the player can choose one and only one player to attack. The probability of killing the chosen one is P.
4): When a player is killed, his role should be announced.

Part IV--How to analysis player's behaviors:

1): The Lord is always be considered as Lord.

2): A player will be considered as Rebel when he attack the Lord.
3): A player will be considered as Rebel when he attack one which considered as Minister before all Rebels are eliminated.
4): A player will be considered as Minister when he attack one which considered as Rebel.
5): A player will be considered as Traitor when he have been considered as Minister and Rebel.
6): The players are not clever enough, they can't analysis other behaviors.

Part V--The attack strategy of each role:

Eliminated player couldn't do anything.
No one will attack eliminated player or himself.
For each strategy, all eligible player have equal probability of being attacked.
The Rebel will choose one of the following strategies by order:
1)The Lord or Ministers if some players are considered as Minister.
2)The Lord or Traitor if a player is considered as Traitor.
3)The Lord.

The Lord and Ministers will choose one of the following strategies by order:
1)The player which be considered as Rebel.
2)The player which be considered as Traitor.
3)If all rebel roles are eliminated, then attack anyone except Lord.
4)The player with unknown role.

The Traitor will choose one of the following strategies by order:
1)The player which be considered as Minister if alive Minister number more then alive Rebel number.
2)The player which be considered as Rebel if alive Rebel number more then or equal alive Minister number.
3)Anyone expect the Lord.
4)The Lord.

Such information will be given:
1): The number of players.
2): Each player's identify.
3): The probability of killing the chosen one.
You should work out the probability to win in 10 round for each role.

Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.
Each case consists of three lines, the first line is an integer N(4<=N<=8), indicating the number of players.
The second line contains N integers, role0,role1,role2...rolen-1, indicating the role of each player.(The first player will always be the Lord).(0, 1, 2, 3 indicating Lord, Minister, Rebel, and Traitor respectively.)
Then the third line contains a real number P(0<P<1).

Output

For each case, output the number of case and the win probability of the 4 roles.Answers are rounded to 3 numbers after the decimal

point.(as shown in the sample output)

Sample Input

```
2
4
0 1 2 3
0.5
4
0 3 2 1
0.5
```

Sample Output

```
Case 1:
Lord and Minister:0.549
Rebel:0.334
Traitor:0.118
Case 2:
Lord and Minister:0.501
Rebel:0.334
Traitor:0.166
```

1007 Street Fighter

Time Limit: 30000/10000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Street Fighter is a popular Fighting game. In the game, you could choose many roles to fight.
Each role has some models, in different model the role has different super skill. Such as Ryu, he has two model -- "Metsu Hadoken" and "Metsu Shoryuken". In "Metsu Hadoken" model, Ryu could beat Chun-Li easyly and in "Metsu Shoryuken" he could beat Ken.

Giving the information of which role in which model could beat which role in which model. Your task is choosing minimum roles in certain model to beat other roles in any model.(each role could only be chosen once)

Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.
The first line of each case contains a integers N(3<=N<=25), indication the number of roles. (roles numbered from 0 to N - 1).
Then N blocks follow, each block contain the information of a role.
The first of each block contains a integer M(1<=M<=2), indication the number of model of this role.(models numbered from 0 to M - 1)

Then M lines follow, each line contain a number K(1<=K<=10), then K pairs integers(role id and model id) follow, indicating the current role in this model could beat that role in such model.

Output

For each case, output the number of case and the minimum roles have to choose to reach the goal.(as shown in the sample output)

Sample Input

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

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

Sample Output

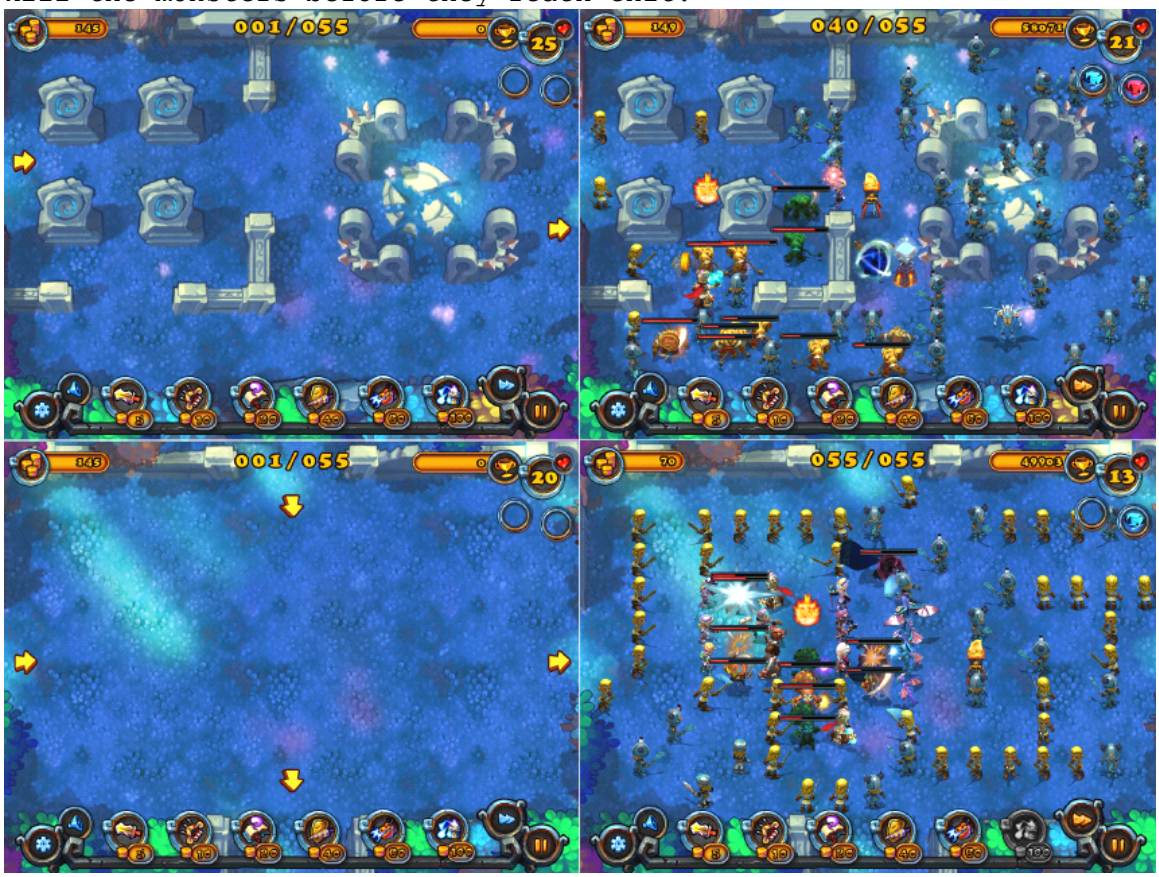
```
Case 1: 2
Case 2: 2
```

1008 Tower Defence

Time Limit: 30000/10000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**
Special Judge

Problem Description

Tower Defence is a popular classic game. During the game, there will appear some monsters at the entry and each will find a shortest path to the exit. Your task is puting some warriors on the map to kill the monsters before they reach exit.



You could put warrior(infinite) on the map to change the pathes of monsters. Of course, the longer the path is, the more advantageous to you.
Now, giving you the information of the map, how do you put the warriors to make the monsters move on the longest path?
The meaning of each character in the map:
'S' represents the entry of the map, one and only one;
'T' represents the exit of the map, one and only one;
'.' represents the grid that monster could move on;
'B' represents the barrier that monster couldn't move on;
'W' represents the warrior that you put.(could only put on '.')

Input

The first line is a number T(1<=T<=30), represents the number of case. The next T blocks follow each indicates a case.

The first line of each case contains two integers N, M (3<=N<=20 , 3<=M<=9)
Then N lines follow, each line contains M chars represent the map, include 'S','T','.','B'.
I promise there must be at least one way from entry to exit.

Output

For each case, first line output the number of case with the longest distance from S to T.(as shown in the sample output)

Then output any one of these ways, using the same format for the map as in the input.
Output a blank line after each case.(Special judge.If you not output a blank line after each case, you will get Wrong Answer)

Sample Input

```
4
3 3
S..
...
..T

5 5
S....
.B.B.
.....
.B.B.
....T

5 5
S...T
..B..
..B..
..B..
.....

5 5
S...T
..B..
..B..
..B..
..B..
```

Sample Output

```
Case 1: 5
T..
...
..S

Case 2: 17
S....
WBWB.
.....
.BWBW
```

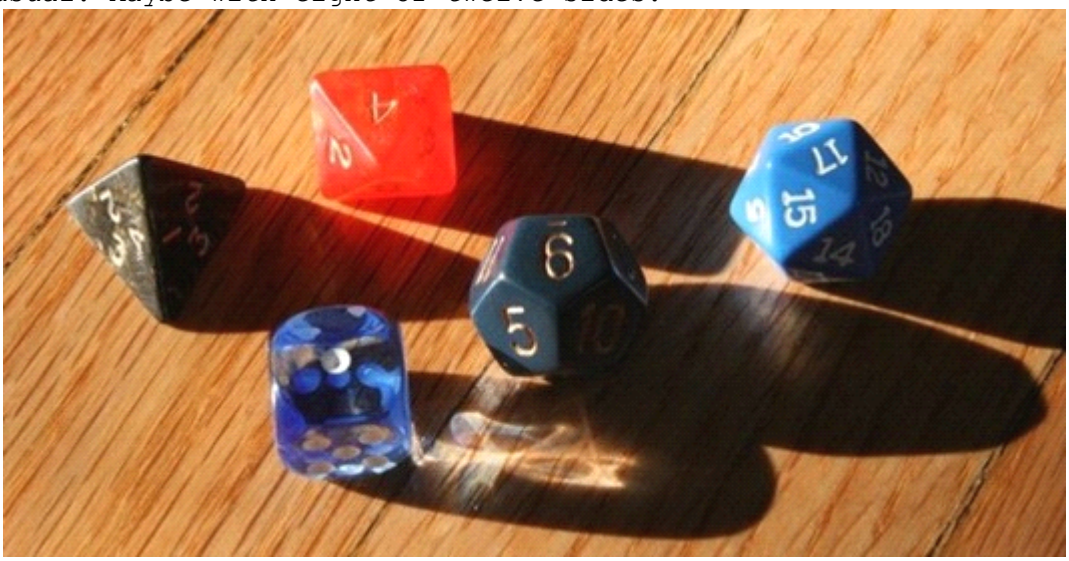
```
....T

Case 3: 17
S.W.T
W.B.W
..B..
.WBW.
.....

Case 4: 5
S...T
..B..
..B..
..B..
..B..
```

1009 Board Game Dice

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

hh is a Board Game hobbyist, he often plays Board Game such as ***Catan***, ***Carcassonne***, ***The Werewolves***, ***A song of ice and fire*** with friends.
To play the games, we need some dices, and these dices are very unusual. Maybe with eight or twelve sides.



hh plays with N friends today(including himself). They'll choose one person to be the judge. But the problem is: there is only a M-sided dice. How to pick a judge with the dice, so that everyone has equal probability of being chosen (the probability should always be 1/N)?
hh has an idea here:
1)Get x
Decide rolling the dice x times to make M^x larger than or equal to N.

2)Players choose sequences
Each player chooses a sequence with x elements (1~M).
For example, a 6-sides dice and x equal to 3, hh will gets sequence 4 5 6. Players' sequences should be different from each other.

3)Pick the judge
Roll the dice for x times, we can get a result sequence, if someone has the same sequence as the result, he will be the judge; otherwise, repeat 1)-3), until the judge is chosen.

It's a bigger project, hh wants know the expected number of times we will need to throw dice to determine the judge.

Input

The first line is a number T(1<=T<=30), which represents the number of cases. The next T blocks following are the cases.
Each case contains two integer N , M(2<=N<=10^9, 2<=M<=20)

Output

For each case, output the number of case and expected number of throw as an irreducible fraction in the following form: "a/b" (as shown in the sample output)

Sample Input

```
2
3 2
9 3
```

Sample Output

```
Case 1: 8/3
Case 2: 2/1
```

1010 World of Warcraft

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

World of Warcraft is a MMORPG developed by Blizzard. When you fight in this game, you should use as many as possible skills to defeat the BOSS.

To defeat the big BOSS - Lich King, you should warm up your fingers (even toes), because you have to press many hotkeys in a second. There are 26 single hotkeys: A~Z, and 78 combination hotkeys: Shift/Ctrl/Alt + A~Z. Each hotkey represents a skill. Single hotkey requires one finger to press. Combination hotkey requires two fingers to press Shift/Ctrl/Alt and A~Z respectively at the same time. The Shift/Ctrl/Alt could only combine with one hotkey in a second, that means if you press Alt/Shift/A/B at the same time, you will use the skill Shift+A and Alt+B or Shift+B and Alt+A as your wish.

The team leader always gives you a list of skills to use. It tells you when to use each skill. In the list, each line contains three integers: si, ei, ki and the hotkey. To win the battle, you must press this hotkey ki times from the si second to ei second. Otherwise, you will be killed by the Lich King.

A finger can only press one key during a second but two fingers couldn't press a same key during a second. The number of your fingers is limited but there are so many skills to use. The question is: is it possible to arrange your finger to execute all of skills in the list?

Input

The first line is a number T(1<=T<=100), represents the number of case. The next T blocks follow each indicates a case: the list given by your team leader.

The first line of each case contains two integer N, K(1<=N<=100 , 1<=K<=20), indicating the number of skills and the fingers(or toes) you could use.

Then N lines follow, each line contains three integers: si, ei, ki(1<=si<=ei<=100,1<=Ki<=ei-si+1) and the hotkey.

It is guaranteed that all the hotkeys are different.

Output

For each case, output the number of case and "Yes" or "No", indicating that whether it is possible to execute all the skills.(as shown in the sample output)

Sample Input

```
2
2 2
1 100 2 Shift + A
2 99 98 Ctrl + B
3 20
1 100 97 A
10 20 10 Shift + B
15 25 10 Shift + C
```

Sample Output

```
Case 1: Yes
Case 2: No
```