Submissions

Scaled Triangle
- 9pt Not attempted **83/100 users** correct (83%)
- 13pt Not attempted **78/87 users** correct (90%)

Painting a Fence
- 7pt Not attempted **190/199 users** correct (95%)
- 13pt Not attempted **113/144 users** correct (78%)

Rainbow Trees
- 9pt Not attempted **71/90 users** correct (79%)
- 15pt Not attempted **68/72 users** correct (94%)

Bus Stops
- 8pt Not attempted **51/57 users** correct (89%)
- 26pt Not attempted **16/23 users** correct (70%)

Top Scores

bmerry	100
dzhulgakov	100
gawry	100
dgozman	100
halyavin	100
pashka	100
mystic	100
Klinck	80
.Invader	76
DmitryKlenov	76

Problem A. **Scaled Triangle**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 9 points	Solve A-small
Large input 13 points	Solve A-large

Problem

You are given two triangle-shaped pictures. The second picture is a possibly translated, rotated and scaled version of the first. The two triangles are placed on the table, with the second one placed completely inside (possibly touching the boundary of) the first one. The second triangle is always scaled by a factor that is strictly between 0 and 1.

You need to process the picture, and for that you need a point in the picture which overlaps with the same point of the scaled picture. If there is more than one solution, you can return any of them. If there are no solutions, print "No Solution" (without the quotes) for that test case.

Input

The first line of input gives the number of cases, **N**. Then for each test case, there will be two lines, each containing six space-separated integers -- the coordinates of one of the triangles -- in the format "x_1 y_1 x_2 y_2 x_3 y_3". The point (x_1, y_1) in the first triangle corresponds to the same corner of the picture as (x_1, y_1) in the second triangle, and similarly for (x_2, y_2) and (x_3, y_3).

Output

For each test case, output one line containing "Case #**x**: " followed two real numbers representing the coordinates of the overlapping point separated by one space character, or the string "No Solution". Answers with a relative or absolute error of at most 10^{-5} will be considered correct.

Limits

1 <= **N** <= 10.
The coordinates of the points will be integer numbers between -10 000 and 10 000. The three points in each triangle will not be collinear.

Small dataset

All tests will contain isosceles right-angle triangles. (i.e., the triangle's angles will be 45 degrees, 45 degrees, and 90 degrees.)

Large dataset

The triangles can have any shape.

Sample input

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

Sample output

```
Case #1: 0.000000 0.000000
Case #2: 2.692308 1.538462
```

Problem B. Painting a Fence

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 7 points	Solve B-small
Large input 13 points	Solve B-large

Problem

You need to hire some people to paint a fence. The fence is composed of 10000 contiguous sections, numbered from 1 to 10000.

You get some offers from painters to help paint the fence. Each painter offers to paint a contiguous subset of fence sections in a particular color. You need to accept a set of the offers, such that:

- Each section of the fence is painted.
- At most 3 colors are used to paint the fence.

If it is possible to satisfy these two requirements, find the minimum number of offers that you must accept.

Input

- One line containing an integer **T**, the number of test cases in the input file.

For each test case, there will be:

- One line containing an integer **N**, the number of offers.
- **N** lines, one for each offer, each containing "**C A B**" where **C** is the color, which is an uppercase string of up to 10 letters, **A** is the first section and **B** is the last section to be painted. 1 ≤ **A** ≤ **B** ≤ 10000.

Output

- **T** lines, one for each test case in the order they occur in the input file, each containing the string "Case #**X**: **Y**", where **X** is the case number, and **Y** is the number of offers that need to be accepted, or "Case #**X**: IMPOSSIBLE" if there is no acceptable set of offers.

Limits

1 ≤ **T** ≤ 50

Small dataset

1 ≤ **N** ≤ 10

Large dataset

1 ≤ **N** ≤ 300

Sample

Input

```
5
2
BLUE 1 5000
RED 5001 10000
3
BLUE 1 6000
RED 2000 8000
WHITE 7000 10000
4
BLUE 1 3000
RED 2000 5000
ORANGE 4000 8000
GREEN 7000 10000
2
BLUE 1 4000
RED 4002 10000
3
BLUE 1 6000
RED 4000 10000
ORANGE 3000 8000
```

Output

```
Case #1: 2
Case #2: 3
Case #3: IMPOSSIBLE
Case #4: IMPOSSIBLE
Case #5: 2
```

In the first test case, accepting both offers will exactly paint the whole fence, 5000 sections each, with no overlap.

In the second case, the painters will overlap, which is acceptable.

In the third case, accepting all four offers would cover the whole fence, but it would use 4 different colours, so this is not acceptable.

In the fourth case, section 4001 cannot be painted.

In the fifth case, we can accept just the first and second offer and successfully paint the fence.

EMEA Semifinal 2008

A. Scaled Triangle
B. Painting a Fence
C. Rainbow Trees
D. Bus Stops

Contest Analysis

Questions asked 2

Submissions

Scaled Triangle

9pt Not attempted **83/100 users** correct (83%)

13pt Not attempted **78/87 users** correct (90%)

Painting a Fence

7pt Not attempted **190/199 users** correct (95%)

13pt Not attempted **113/144 users** correct (78%)

Rainbow Trees

9pt Not attempted **71/90 users** correct (79%)

15pt Not attempted **68/72 users** correct (94%)

Bus Stops

8pt Not attempted **51/57 users** correct (89%)

26pt Not attempted **16/23 users** correct (70%)

Top Scores

bmerry	100
dzhulgakov	100
gawry	100
dgozman	100
halyavin	100
pashka	100
mystic	100
Klinck	80
.Invader	76
DmitryKlenov	76

Problem C. Rainbow Trees

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 9 points	Solve C-small
Large input 15 points	Solve C-large

Problem

In graph theory, a *tree* is a connected, undirected simple graph with no cycles. A tree with **n** nodes always has **n** - 1 edges.

A *path* in a tree is a sequence of distinct edges which are connected (each pair of consecutive edges in the path share a vertex).

Consider a tree with **n** vertices and **n**-1 edges. You can color each edge in one of **k** colors.

An assignment of colors to edges is a *rainbow coloring* if in every path of 2 or 3 edges, the colors of the edges are different. (i.e., every two consecutive edges have different colors, and every three consecutive edges have different colors).

Given a tree and the number of colors **k**, find the number of rainbow colorings modulo 1000000009.

Input

The first line of input gives the number of test cases, **C**. Then for each of the **C** cases, there will be:

- One line containing two integers in the format "**n k**". **n** is the number of nodes in the tree, and **k** is the number of colors available.
- **n** - 1 lines, one for each edge, containing two integers "**x y**", indicating that the edge is between node **x** and node **y**. Nodes are numbered from 1 to **n**.

Output

For each test case, output one line. That line should contain "Case #**X**: **Y**", where **X** is 1-based number of the case, and **Y** is the answer for that test case.

Limits

1 <= **k** <= 1000000000
All the node numbers are between 1 and **n**, inclusive.

Small dataset

1 <= **C** <= 100
2 <= **n** <= 20

Large dataset

1 <= **C** <= 40
2 <= **n** <= 500

Sample

Input	Output
2 4 10 1 2 1 3 1 4 5 3 1 2 2 3 3 4 4 5	Case #1: 720 Case #2: 6

In the first case, the tree has four nodes. There are edges from one node to each of the other three. Each pair of these edges are adjacent, so for there to be a rainbow coloring, all the edges must have different colors. There are therefore 10 x 9 x 8 = 720 rainbow colorings.

In the second case, the tree itself is a path of 4 edges, and there are 3 colors.

The first three edges must all have different colors, so there are 3 x 2 x 1 colorings for these, and then there is only one choice for the fourth edge, so there are 6 rainbow colorings.

Submissions

Scaled Triangle

9pt	Not attempted **83/100 users** correct (83%)
13pt	Not attempted **78/87 users** correct (90%)

Painting a Fence

7pt	Not attempted **190/199 users** correct (95%)
13pt	Not attempted **113/144 users** correct (78%)

Rainbow Trees

9pt	Not attempted **71/90 users** correct (79%)
15pt	Not attempted **68/72 users** correct (94%)

Bus Stops

8pt	Not attempted **51/57 users** correct (89%)
26pt	Not attempted **16/23 users** correct (70%)

Top Scores

bmerry	100
dzhulgakov	100
gawry	100
dgozman	100
halyavin	100
pashka	100
mystic	100
Klinck	80
.Invader	76
DmitryKlenov	76

Problem D. Bus Stops

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 8 points	Solve D-small
Large input 26 points	Solve D-large

Problem

In the First City of Mars there are **N** bus stops, all aligned in a straight line of length **N**-1 km. The mayor likes to keeps things simple, so he gave the bus stops numbers from 1 to **N**, and separated adjacent stops by exactly 1 km.

There are also **K** buses in the city. The mayor has to plan the bus schedule and he would like to know in how many ways that can be done. This number can be very large. Luckily there are a few constraints:

- In the beginning of the day all the buses are in the first **K** bus stops (one bus per stop)
- Buses only move from the left to the right (1 is the leftmost bus stop)
- At the end of the day all the buses must be in the last **K** bus stops (one bus per stop)
- In each bus station exactly one bus has to stop
- For the same bus the distance between any two consecutive stops is at most **P** km

Help the mayor evaluate the number of schedules. However try not to give him very bad news (a lot of schedules) so just output the real number modulo 30031.

Input

The first line in the input file is the number of cases **T**.
Each of the next **T** lines contains 3 integers separated by one space: **N**, **K** and **P**.

Output

For each case output the number of ways to plan the bus schedules (modulo 30031) in the format "Case #**t**: [number of ways modulo 30031]" where **t** is the number of the test case, starting from 1.

Limits

1 < **T** ≤ 30
1 < **P** ≤ 10
K < **N**
1 < **K** ≤ **P**

Small dataset

1 < **N** < 1000

Large dataset

1 < **N** < 10^9

Sample

Input	Output

`3`
`10 3 3`
`5 2 3`
`40 4 8`
	`Case #1: 1`
	`Case #2: 3`
	`Case #3: 7380`

Let's name the buses: A, B, C...
For the first case there is only one possible way of planning the schedule: A → 1, 4, 7, 10. B → 2, 5, 8. C → 3, 6, 9.
For the second case the possible ways of planning are:
(A → 1,3,5. B → 2,4),
(A → 1,3,4. B → 2,5),
(A → 1,4. B → 2,3,5).