CHN01

Malvika conducts her own ACM-ICPC contest series

After winning numerous programming competitions, Malvika has got bored of participating, and is thinking of holding her own series of contests, named "Advanced Contests by Malvika: India's Coolest Programming Contests (ACM-ICPC)". Each contest will have three problems, one each of easy, medium and hard levels.
Now, after some time, Malvika realized that creating hard problems for a contest is not as easy for her as solving them. So she asked her friend Animesh to join the team and help her with it. Animesh himself is no good with creating problems, but he suggested her a clever strategy to reduce the number of hard problems she needs to create. He suggested that ACM-ICPC could have three types of contests, comprising of the following problem levels:

- Type 1: 1 easy, 1 medium, 1 hard
- Type 2: 1 easy, 2 medium
- Type 3: 2 easy, 1 medium

One day, Malvika and Animesh sat down and prepared **e** easy problems, **m** medium, and **h** hard problems. As we know that it is harder to make higher difficulty problems, the number of hard problems will be no more than medium and similarly, the number of medium level problems, will be no more than easy level.

Now, they want to use the above problems to organizes a series of contests. Also, as they want the series to be interesting, so they don't want to have two consecutive contests in the series to be of the same type.

Can you please help them in finding out the maximum number of contests they can conduct in the series?

Input

- The first line of input contains a single integer **T** denoting the number of test cases.
- The only line of each test case contains three space separated integers — **e, m, h** — as defined in the statement.

Output

- For each test case, output a single integer in a line, the answer to the problem.

Constraints

- $1 \leq \mathbf{T} \leq \mathbf{10^5}$
- $0 \leq \mathbf{e, m, h} \leq \mathbf{10^5}$
- $\mathbf{h} \leq \mathbf{m} \leq \mathbf{e}$
- Sum of variables **e** for all the test cases won't exceed $\mathbf{10^7}$.

Example

Input:
2
1 1 1
4 4 1

Output:
1
3

Explanation

In the first example, they have 1 easy, 1 medium and 1 hard problem. They can conduct one contest of type 1. This is the maximum number of contests they can conduct.

In the second example, they have 4 easy, 4 medium, and 4 hard problems. They can conduct 3 contests, one each of type 1, 2, and 3, which require a total of 4 easy, 4 med and 1 hard problems. This is the maximum number of contests they can conduct. Hence, the answer is 3.

CHN02

Animesh decides to settle down

After trying his luck in programming contests, Animesh has decided to accept his fate and settle down the good old-fashioned way. He has decided to own a farm and be a shepherd. He knows of **n** grass varieties, each of which grows in a convex polygon area. Some of these polygons may intersect each other. Each grass is lush green and nutritious, and he cannot make up his mind on where he should settle.

His friend Malvika, a long time shepherd herself (this is *the other* Malvika), suggested that he buy a piece of land where he can find all the **n** grass varieties, so that his sheep will be fat and produce lots of wool. Animesh liked the idea and wants to buy as much of such a pasture as possible. However, being short on money, he decided to be judicious, and only buy land which is covered by all the **n** grasses. That is, he would not buy even a bit of land which isn't part of all **n** of the grassy polygons. Can you please help in finding out the maximum area of land he can buy?

Input

- First line will contain an integer **n**, denoting the number of polygons.
- Then, the next lines will contain description of **n** polygons, description of the **i**th polygon is given as below.
  - First line will contains a single integer **c** denoting number of points in the polygon.
  - each of the next **c** lines will have two space separated integers **x[j], y[j]** denoting **x** and **y** coordinate respectively of the **j**-th point of the polygon. These points will be given in anti-clockwise order.

Output

- Output your answer in a separate line. Your answer will be considered correct if it has an absolute error less than **1e-2**.

Constraints

- **$1 \le n \le 30$**
- **$3 \le c \le 30$**
- **$-10^5 \le x, y \le 10^5$**

Example

Input:
3
4
15 0
15 5
0 5
0 0
4
2 3
15 3
15 10
2 10
4
19 7
5 7
5 -1
19 -1

Output:
20.000

Explanation

Each region in the current case is rectangular in shape. You can check that the area of intersection of all the three input rectangles is another rectangle with corners at **(5,3), (15,3), (15,5)** and **(5,5)**. This is the place Animesh is going to settle in. The area of this rectangle is 10 * 2 = 20.

CHN03

Animesh practices some programming contests

As you know, Malvika has created some **n** programming contests. Each of the contests has three problems, categorized as easy, medium and hard on difficulty level. For the **i**th contest, easy problem takes **TE[i]** hours and gives you **PE[i]** pleasure. Similarly, medium problem takes **TM[i]** hours, gives **PM[i]** pleasure, and a hard one has the values **TH[i]**, **PH[i]**.

Today, Animesh wanted to practice some of them. Animesh has a really bad habit of trying problems for only a few minutes and saying to Malvika "I am a noob, you are a pro. It's some weird shit I don't know. Please, tell me the solution, bro!" Having been irritated by this behaviour numerous times in the past, she granted him **K** special powers he can use before starting the practice session. By using a single power, Animesh can pick any two problems irrespective of their difficulty from two different contests and swap them.

Animesh has at max **time** hours before he gets frustrated by his noobness and ends the practice session. He wants to make the maximum use of it by getting as much pleasure out of this activity as possible. Animesh also gets bored with the contest themes fairly quickly, so he does not want to solve more than one problem from any contest. Can you help Animesh in estimating the maximum amount of pleasure he can achieve out of this activity?

Input

The first line of input contains a single integer **T** denoting number of test cases

For each test case :

- The first line contains three space separated integers **n, k, time**.
- Each of the next **n** lines contain three space separated integers denoting **TE[i], TM[i], TH[i]**.

- Each of the next **n** lines contain three space separated integers denoting **PE[i], PM[i], PH[i]**.

Output

- For each test case, output a single integer in a separate line corresponding to the answer of the problem.

Constraints

- 1 ≤ **T** ≤ **10**
- 1 ≤ **n** ≤ **50**
- 0 ≤ **k** ≤ **n * n**
- 1 ≤ **time** ≤ **50**
- 1 ≤ **TE[i], TM[i], TH[i]** ≤ **50**
- 1 ≤ **PE[i], PM[i], PH[i]** ≤ **100000**

Example

Input:
2
1 0 5
1 2 3
5 3 6
2 1 6
1 2 1
2 3 3
5 3 3
4 6 5

Output:
6
11

Explanation

In the first example, Malvika has prepared only one programming contest, its easy problem takes 1 hour and gives 5 units of pleasure. The medium one takes 2 hours and gives 3 units of pleasure whereas the same values for hard are 3 hours and 6 units of pleasure. Animesh has only 5 hours and can select at max one problems out of these to solve. The hard problem is the best candidate to choose for him, it will him give him 6 units of pleasure and take his 5 hours. So, the answer is 6.

In the second example, Malvika has prepared two programming contests.

- First programming contest:
  - Easy problem — 1 hour and 2 units of pleasure.
  - Medium problem — 2 hours and 3 units of pleasure.
  - Hard problem — 1 hours and 3 units of pleasure.
- Second programming contest:
  - Easy problem — 2 hours and 4 units of pleasure.
  - Medium problem — 3 hours and 6 units of pleasure.
  - Hard problem — 3 hours and 5 units of pleasure.

Animesh has 6 hours and can use at max one swapping power. He can use the special power to swap the medium problem of second contest with any problem of first contest. After doing that, he can select the two problems taking 3 hours and giving pleasure of 6 and 5 units to get a total pleasure of 11 units.

CHN04

Malvika gets bored along with her Cats

Malvika's cats, Chingam and Jimma are also bored of solving problems. Fortunately, Chingam has found **n** bars of dark chocolate. They all have the same width, but differ in their lengths. In fact, no two chocolate bars have the same length. All the **n** bars are kept in a row. Let's say that their lengths are given by **L[1], L[2], ... , L[n]** from left to right. That is, length of bar **i** is **L[i]**.

Chingam decided to play a little game with Jimma. In one step, Jimma has to choose some bar which has not been eaten yet. Chingam hates inversions, so he will instantaneously eat that bar, along with every other Bar with which it forms an **Inversion Pair**. We say that bars **i** and **j** form an **Inversion Pair**, if the bar to the left is longer than the bar to the right.

That is, if Jimma selects bar **i**, whose length is **L[i]**, then Chingam will eat this, as well as every bar **j** such that **j > i** and **L[j] < L[i]**, as well as every bar **k** such that **k < i** and **L[k] > L[i]**. After this, the step ends, they proceed to the next step, where Jimma selects another bar and the whole process repeats. The game ends when all the **n** chocolate bars are eaten.

Jimma doesn't like this game because she does not get any chocolate. So, she wants to finish it as soon as possible and go back to sleep. Please help her find the minimum number of steps needed for the game to end.

Input

- The first line of input contains a single integer **T** denoting number of test cases
- The first line of each test case contains one integer: **n**.
- The second line contains **n** space separated integers, which correspond to **L[1], L[2], ... , L[n]**.

Output

- For each test case, output a single integer in a line corresponding to the answer of the problem.

Constraints

- 1 ≤ **T** ≤ **5**
- 1 ≤ **n** ≤ $\mathbf{10^5}$
- 1 ≤ **L[i]** ≤ **n**

Example

Input:
1
4
3 1 4 2

Output:
2

Explanation

In the first step, Jimma can choose bar 4, which is of length 2. Bar 3 and bar 1 form **Inversion Pairs** with bar 4. So Chingam eats all these 3 bars. In the second step, Jimma chooses bar 2, and she's done. You can check that Jimma cannot get rid of all the four bars in one step.

CHN05

Malvika conducts a Programming Camp

After acquiring an extraordinary amount of knowledge through programming contests, Malvika decided to harness her expertise to train the next generation of Indian programmers. So, she decided to hold a programming camp. In the camp, she held a discussion session for **n** members (**n-1** students, and herself). They are sitting in a line from left to right numbered through **1** to **n**.

Malvika is sitting in the **n**th spot. She wants to teach **m** topics of competitive programming to the students. As the people taking part in the camp are all newbies, they know none of the topics being taught, i.e., initially, the first **n - 1** people in the line know none of the topics, while the **n**th knows all of them.

It takes one hour for a person to learn a topic from his neighbour. Obviously, one person cannot both teach a topic as well as learn one during the same hour. That is, in any particular hour, a person can either teach a topic that he knows to one of his neighbors, or he can learn a topic from one of his neighbors, or he can sit idly. It is also obvious that if person x is learning from person y at a particular hour, then person y must be teaching person x at that hour. Also, note that people can work parallely too, i.e., in the same hour when the 4th person is teaching the 3rd person, the 1st person can also teach the 2nd or learn from 2nd.

Find out the minimum number of hours needed so that each person learns all the **m** topics.

Input

- The first line of input contains a single integer **T** denoting number of test cases.
- The only line of each test case contains two space separated integers **n, m** as defined in the statement.

Output

- For each test case, output a single integer in a line corresponding to the answer of the problem.

Constraints

- 1 ≤ **T, n, m** ≤ **100**

Example

Input:
2
2 1
3 2

Output:
1
4

Explanation

In the first example, there are two people. Second person is Malvika and she has to teach only one topic to the first person. It will take only one hour to do so.

In the second example, there are three people. The 3rd person is Malvika and she has to teach only two topics to 1st and 2nd person. In the 1st hour, she teaches the 1st topic to the 2nd person. Now, in the 2nd hour, the 2nd person will teach the 1st topic to the 1st person. In the 3rd hour, Malvika will teach the 2nd topic to the 2nd person. Now the 2nd person will teach that topic to the 1st in the 4th hour. So, it takes a total of 4 hours for all the people to know all the topics.

CHN06

King Animesh decides to have a voyage to the sun

King Animesh has decided to go on a voyage to the Sun. He has decided to take only one of his courtiers with him. Everyone knows that the king has two favorite courtiers, Arjun and Praveen, and so they are competing between themselves to go with the King. Arjun is a great archer, so he has decided to impress the king by demonstrating his perfect abilities. Praveen is smart, he knows exactly what the King likes - Graphs of Perfection. Graphs of Perfection are weighted complete bipartite graphs with **2*N** vertices, which have a property that **every Perfect Matching in the graph has the same cost**. The cost of a Perfect Matching is just the sum of costs of all the **N** edges in it.

Such graphs are rare, but Praveen had found one in the Forest of Perfection and he had kept it carefully in a box. The **N** nodes in the left column are labelled **{1,2, … ,N}**, and the **N** nodes in the right column are labelled **{N+1, N+2, … , 2*N}**. The cost associated with an edge between node **u** in left column and node **v** in right column, is denoted by $\mathbf{C_{uv}}$.

Unfortunately, when it is Praveen's turn to present the graph to the King, he realizes that his rival Arjun has made a hole in the box and as a result, some of the edges have fallen off. He tries to remain calm and presents the graph to the King claiming that he can assign costs to the missing edges in such a way that the graph becomes a Graph of Perfection again. Praveen knows that the remaining edges have a magical property - **there is a unique way to assign costs to the missing edges such that the graph becomes a Graph of Perfection**. But he doesn't tell that to the King, and so the King gets highly impressed by his claim and tells him to report the sum of the squares of all the $\mathbf{N^2}$ edge costs modulo $\mathbf{10^9+7}$. Alas! Praveen is not *that* smart, but we hope you are! You are given the costs of the edges which are still in the box; please help Praveen find the required sum.

Note:

- A complete bipartite graph means that there is an edge between every node in the left column and every node in the right column.
- A Perfect Matching in this graph is a set of **N** edges such that no two edges among them have any end point in common.

Input

- The first line contains two integers: **N** and **E**.
- The next **E** lines of the test case contain three integers each: **u**, **v** and **x**. This signifies Praveen knows the cost of the edge from vertex **u** in the left column to the vertex **v** in the right column, and that cost is equal to **x**. That is, $\mathbf{C_{uv}} = \mathbf{x}$
- It is guaranteed that there is a unique way to assign costs to the missing edges such that the given graph in the input becomes a Graph of Perfection.

Output

- For each test case, output a single integer in a line, corresponding to the answer of the problem.

Constraints

- $1 \le \mathbf{N} \le \mathbf{10^5}$
- $1 \le \mathbf{E} \le \mathbf{10^6}$
- $1 \le \mathbf{u} \le \mathbf{N}$
- $\mathbf{N+1} \le \mathbf{v} \le \mathbf{2*N}$
- $1 \le \mathbf{x} \le \mathbf{10^5}$, but note that the edge costs which are not in the input need not satisfy this constraint. That is, this constraint does not apply for the missing edges.

Example

Input:
2 3
1 3 2
1 4 3
2 3 5

Output:
74

Explanation

Out of the 4 edges, the costs of 3 edges are already given to us. The edge (2, 4) must cost 6 to satisfy the "All Perfect Matchings should have equal costs" constraint. Therefore, now, the sum of the squares of the costs of all the 4 edges is $(2^2 + 3^2 + 5^2 + 6^2) = (4 + 9 + 25 + 36) = 74$

CHN07

Malvika and Animesh play Red-Blue cards Game

After 2 hours of Mathematics and chill, Malvika and Animesh got bored of solving problems and decided to play a game. They have **n** cards with them, each of which is red on one side and blue on the other. They initially put them in a row, from left to right. They then takes turns, with Malvika going first.

In a single turn, the player should choose a card which has its red face up. The player also chooses zero or more consecutive blue faced cards, with the sequence starting immediately on the right of the chosen red card. All the chosen cards are then flipped. That is, the red faced card will now be blue. The zero or more blue faced cards will now be red. Then the turn ends, and it's the other player's turn.

You will be given the initial arrangement of the **n** cards as a string of R's and B's, which imply red facing-up and blue facing-up, respectively. They will be given in order, from left to right.

The player who is not able to make any move in his/her turn loses. Find out the winner, assuming that both Malvika and Animesh play optimally.

Input

- The first line of input contains a single integer **T** denoting the number of test cases.
- The only line of each test case contains a string of 'R's and 'B's.

Output

- For each test case, output a single line. It should be "Malvika" (without quotes) if Malvika is the winner, and "Animesh" (without quotes) if Animesh is the winner.

Constraints

- 1 ≤ **T** ≤ **100**
- 1 ≤ **n** ≤ **500**, where **n** refers to the length of each string.

Example

Input:
2
R
BB

Output:
Malvika
Animesh

Explanation

In the first example, Malvika will flip the red faced card in the first move, so it will now be blue faced. Now Animesh can't make any move, so he will lose. Hence Malvika is the winner.

In the second example, Malvika can not make any move in the first turn itself, and therefore she loses. Hence Animesh is the winner.

CHN08

Crazy Malvika discovers Crazy Fibonacci function

Malvika was getting bored of the usual Fibonacci problems, and decided to add a little twist to it. She defined a new function **f()** with the following properties:

- She'll give you two integers, **A** and **B**. **f(1)** is defined to be **A** and **f(2)** is **B**.
- And for all integers **x ≥ 2, f(x) = f(x-1) + f(x+1)**.

She'll give an integer **N**, and you have to find out what **f(N)** is. Output the answers modulo **10^9+7**.

Input

- The first line of input contains a single integer **T** denoting number of test cases.
- The only line of each test case contains three integers: **A**, **B** and **N**, denoting **f(1)**, **f(2)** and the query.

Output

- For each test case, output a line which contains a single integer, corresponding to **f(N)** for the given input.

Constraints

- $1 \le \mathbf{T} \le \mathbf{10^5}$
- $\mathbf{-10^9} \le \mathbf{A}, \mathbf{B} \le \mathbf{10^9}$
- $1 \le \mathbf{N} \le \mathbf{10^9}$

Example

Input:
2
10 17 3

23 17 3

Output:
7
1000000001

Explanation

In the first test case, **f(3)** = 7, and so that is the output.

In the second test case, **f(3)** = -6 and the answer modulo $\mathbf{10^9+7}$ is 1000000001.

CHN09

Malvika is peculiar about color of balloons

Little Malvika is very peculiar about colors. On her birthday, her mom wanted to buy balloons for decorating the house. So she asked her about her color preferences. The sophisticated little person that Malvika is, she likes only two colors — amber and brass. Her mom bought **n** balloons, each of which was either amber or brass in color. You are provided this information in a string **s** consisting of characters 'a' and 'b' only, where 'a' denotes that the balloon is amber, where 'b' denotes it being brass colored.

When Malvika saw the balloons, she was furious with anger as she wanted all the balloons of the same color. In her anger, she painted some of the balloons with the opposite color (i.e., she painted some amber ones brass and vice versa) to make all balloons appear to be the same color. As she was very angry, it took her a lot of time to do this, but you can probably show her the right way of doing so, thereby teaching her a lesson to remain calm in difficult situations, by finding out the minimum number of balloons needed to be painted in order to make all of them the same color.

Input

- The first line of input contains a single integer **T**, denoting the number of test cases.
- The first and only line of each test case contains a string **s**.

Output

- For each test case, output a single line containing an integer — the minimum number of flips required.

Constraints

- **1 ≤ T ≤ 100**
- **1 ≤ n ≤ 100**, where **n** denotes to the length of the string **s**.

Example

Input:
3
ab
bb
baaba

Output:
1
0
2

Explanation

In the first example, you can change amber to brass or brass to amber. In both the cases, both the balloons will have same colors. So, you will need to paint 1 balloon. So the answer is 1.

In the second example, As the color of all the balloons is already the same, you don't need to paint any of them. So, the answer is 0.

CHN10

Animesh does not gift Malvika on her birthday

Malvika was sad because Animesh, the miser, did not buy her even a chocolate bar on her birthday. Animesh, feeling sorry about his behavior, repented and asked Malvika how he could make up for this catastrophic mistake. Enraged as she was, Malvika gave him an empty graph **H** of **n** vertices and asked him to add edges into it and convert the graph into a graph that she likes.

Malvika always keeps a graph **G** with **n** vertices and **m** edges with her. Malvika likes a graph **H** if, and only if, she can find a permutation **P** of **1, 2, ... , N** such that there's an edge between vertices **u** and **v** in **H** if and only if there's an edge between vertices **P[u]** and **P[v]** in **G**.

Animesh thought for days and days about how he could add edges into the empty graph **H** so that he can get a graph that Malvika likes. After putting in a lot of effort, he found such a graph and said that this is the graph you desired.

Malvika was still really angry with him and asked *little* Animesh to come up as many graphs as possible, such that Malvika likes them. Given his recent track record, you know this is going to take him at least a few years to solve by himself. Can you help Animesh find out how many such graphs he has to come up with? Please output your answer modulo $\mathbf{10^9 + 7}$.

Two ways of adding edges **A, B** are considered different if there is a an edge **(u, v)** which is present in **A** but not in **B**.

You are given the adjacency matrix of the graph **G** as input. The graph has a special property — each row and column of the graph has atleast **n - 3** 1's.

Note that adjacency matrix of a graph **G** containing **n** vertices is a matrix **A** of size **n * n**. If there is an edge between vertex **i** and **j**, then **A(i, j) = 1**, otherwise **A(i, j) = 0**. In our case, as the graph is undirected, the matrix will be a symmetric matrix. Also, the entries on the diagonal will be all zeros.

Input

The first line of input contains a single integer **T** denoting the number of test cases.

For each test case:

- First line contains an integer **n** denoting the number of vertices in graph **G**.
- Each of the next **n** lines contains **n** characters, each of them will be either '0' or '1'. The **j**th character in the **i**th line denotes **A[i][j]**, where **A** is the adjacency matrix.

Output

- For each test case, output your answer in a separate line.

Constraints

- **1** ≤ T
- n ≤ **50**

Example

Input:
2
2
01
10
3
001
001
110

Output:
1
3

Explanation

In the first example, the graph **G** has an edge between 1 and 2. Animesh has to add only one edge in the graph **H** between vertices 1 and 2 to please her. This is the only graph he has to come up with.

In the second example, the graph **G** has edges {1, 3} and {2, 3}. Animesh can come up with the following graphs {{1, 3}, {2, 3}}, {{2, 1}, {3, 1}}, {{1, 2}, {3, 2}}. Malvika likes all of these graphs. In this representation, {{1, 3}, {2, 3}} means that a graph has 3 vertices and two edges — between 1 and 3, and between 2 and 3.

CHN11

Animesh has a war with tribal leader Malvika

Malvika is the leader of a large tribal army, with soldiers from multiple tribes. There are a total of **n** soldiers in the army. You are given an array **a** of length **n**, where **a[i]** denotes the tribe number of the **i**th soldier.

Now Animesh, the leader of the neighboring tribe and staunch enemy of Malvika, has attacked her army and declared a war which would later be regarded as the greatest tribal war of the contemporary world. For this soon-to-be-lengendry war, Malvika wanted to evaluate the strength of her army. Malvika's army's strength is based on love, i.e., the pairwise strength of two people from the different tribes would be zero, while for those from the same tribe, it will be the distance between their positions in the line. The more the distance, the lonelier they feel. The lonelier they feel, the more they love each other. As their love grows, so does the strength of the army.

Formally, strength of an army denoted by an array **a** will be the sum of pairwise strengths of each pair of people from the same tribe, i.e. sum of all **|i - j|** for **1 ≤ i ≤ j ≤ n** such that **a[i] = a[j]**.

Malvika wants to rearrange her army to maximize the strength. Let **M** denote the maximum strength of her army after rearranging the soldiers. You have to help her find out the number of rearrangements of her army having the maximum strength. The more the rearrangements are, there are more chances of her winning the battle. As they say, you can win the battle with a lot of strength, but you need rearrangements of that strength to win the war. Since the battle is expected to be long drawn, the answer could be very large. So, print your answer modulo **10^9 + 7**.

Two arrangements of the army — **A** and **B** — are said to be distinct if there is an index **i** from **1** to **n**, such that **A[i] ≠ B[i]**. That is, if some position in the army is occupied by soldiers from different tribes in the two arrangements.

Input

- The first line of input contains a single integer **T** denoting the number of test cases.
- The first line of each test case contains a single integer **n** as defined in the statement.
- The second line of each test case will contain **n** space separated integers denoting the array **a**.

Output

- For each test case, output a single integer in a line corresponding to the answer of the problem.

Constraints

- $1 \le T \le 10$
- $1 \le n \le 10^5$
- $1 \le a[i] \le 10^9$

Example

Input:

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

Output:

```
2
4
1
```

Explanation

In the first example, the strength of the army is zero irrespective of the way she arranges her army. There are two ways of arranging the army, namely {1, 2} and {2, 1}.

In the second example, she can arrange a maximum strength army in four ways. Those are {1, 2, 1, 2}, {1, 2, 2, 1}, {2, 1, 1, 2}, {2, 1, 2, 1}. Strength of her army in all the four cases will be 4, the maximum possible strength of the army that she can get.

CHN12

Praveen falls from a tall tree

Praveen has climbed a tall tree and now he can't get down! While he is waiting for Arjun to bring a ladder, he has decided to amuse himself by numbering the **N** nodes of the tree from from **1** to **N** and associating a value **S[i]** with every vertex.

For that, he applies the following procedure.

val = 1
Let T be our tree.
while T is not empty:
- Identify the branching nodes of tree T. A node of tree T is said to be a branching node if its degree > 2.
- Choose all the nodes of T which have a path to any leaf node not passing through any of the branching nodes.
- Remove all of these chosen nodes from the tree T.
  - Set S value of all these removed nodes to be val.
- Increase val by 1, i.e. val += 1

Note that first step of identification of branching nodes is re-done in each execution of the while loop.

Please check the example problem statement to understand how this process works. Note that Praveen cannot actually remove the nodes from the tree. He just simulates the procedure in his head's supercomputer.

After Arjun comes back with the ladder, Praveen decides to ask him **Q** queries about the tree. Each query will contain two nodes **u, v**. Let the path in the tree from node **u** to **v**, be **u, u_1, u_2, .., u_r, v**. Consider the array **S[u], S[u_1],..,S[u_r], S[v]**. For each query, Arjun has to find the number of inversion pairs in this array. **i** and **j** form an inversion pair, if **i > j** and **S[i] < S[j]**. All these hours sitting atop the tree have made Praveen light in the head, and he refuses to come down until Arjun answers all his

queries.
Unfortunately, Arjun does not have a supercomputer in his head like Praveen does and he has also left his laptop at home. So please help Arjun answer these queries and get Praveen down.

Input

The first line of input contains a single integer **T** denoting the number of test cases.

For each test case:

- The first line contains two space separated integers **N, Q** as defined in the statement
- Each of next **N - 1** lines contains two space separated integers **u, v** denoting that there is an edge between vertex **u** and **v** in the tree.
- Each of next **Q** lines will contain two space separated integers **u, v**, the nodes corresponding to the query.

Output

- For each test case, print **Q** lines corresponding to the answers of the queries in separate lines.

Constraints

- $1 \le T \le 10^5$
- $1 \le N, Q \le 10^5$
- $1 \le u, v \le N$
- Sum of all **N**, as well as sum of all **Q**, won't exceed 10^5.

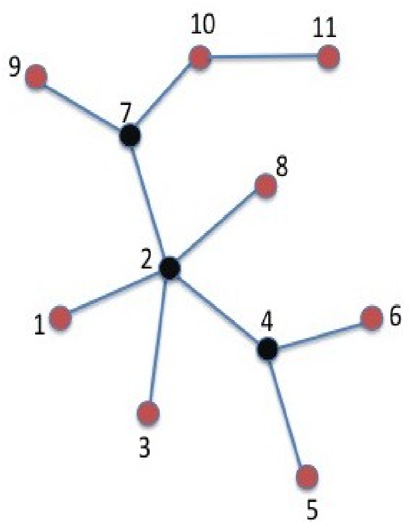
Example

Input 1:
1
11 6
1 2
2 3
2 4
4 5
4 6

2 8
2 7
7 9
7 10
11 10
5 10
1 9
6 10
3 7
2 9
9 2

Output 1:
3
2
3
0
2
0

Explanation



The figure is the tree corresponding to the given sample input.

The nodes marked red have S value 1 and the black nodes have S Value 2. In the first step, the branching nodes are 2, 4 and 7. Nodes 1, 3, 5, 6, 8, 9, 11 are leaf nodes and can be removed, i.e they have S value 1. Even 10 can be removed and it also has S value 1. The branching nodes cannot be touched. In the next iteration, with val = 2, there are no branching nodes and we can remove nodes 2, 4 and 7 giving them S Value 2.

The first query is (5,10). The path between nodes numbered 5 and 10 passes through nodes {5,4,2,7,10} with S values {1,2,2,2,1}. The number of inversions in this array is 3 formed by 3 pairs of {2,1}.

The second query is (1,9). The path between nodes numbered 1 and 9 passes through nodes {1,2,7,9} with S values {1,2,2,1}. The number of inversions in this array is 2 formed by 2 pairs of {2,1}.

The third query is (6,10). The path between nodes numbered 6 and 10 passes through nodes {6,4,2,7,10} with S values {1,2,2,2,1}. The number of inversions in this array is 3 formed by 3 pairs of {2,1}.

The fourth query is (3,7). The path between nodes numbered 3 and 7 passes through nodes {3,2,7} with S values {1,2,2}. There are no inversions in this array.

The fifth query is (2,9). The path between nodes numbered 2 and 9 passes through nodes {2,7,9} with S values {2,2,1}. The number of inversions in this array is 2 formed by 2 pairs of {2,1}.

The last query is (9,2). The path between nodes numbered 9 and 2 passes through nodes {9,7,2} with S values {1,2,2}. The number of inversions in this array is zero.

Input 2:
1
20 4
1 2
2 3
2 4
4 6
5 6
6 20

8 20
7 20
4 9
9 10
10 19
17 19
18 19
10 11
11 12
11 13
13 14
13 15
15 16
4 10
1 16
17 18
11 11

Output 2:

0
18
1
0