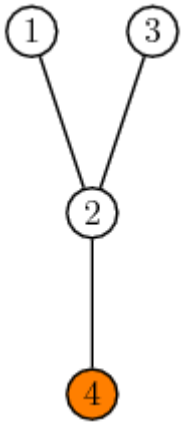
Problem A. Play with [a] Tree

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Hey, ACRush and Jelly are playing a game ! Let take a look at its rule:
You are given a tree. Two players take turns cutting edges on a tree. Some nodes is on the "ground". When a player cuts an edge, all the edges that are no longer connected to the ground disappear. The player who can not take a move loses.

ACRush plays first. Both of them are very good players. If you know state of the tree they are playing with, can you guess who will win?



Node 4 is on the ground.

Input

Input consists of multiple test-cases. The first line contains one integer t - number of cases ($0 < t \leq 20$).
For each case, the input format is following.
The first line contains one integer N ($1 \leq N \leq 100000$). The next line N integers s_i (1 or 0). If s_i is 1, the i^{th} node is on the ground. If s_i is 0, the i^{th} node is not on the ground.

Each line of the following $N-1$ lines contains two integers u, v. They denote there is an edge between node u and node v ($1 \leq u, v \leq N$).
There is no blank line after each case.

Output

For each case, output who will win the game. If ACRush wins, output 1; otherwise, output 0 (Jelly wins).
There is no blank line after each case.

Sample input and output

Standard Input	Standard Output
1 4 0 0 0 1 1 2 2 3 2 4	1

Problem B. The easiest problem? (Yes/No)

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	0.1-10 seconds
Memory limit:	256 megabytes

You are given an unweighted, undirected tree T. We say T is special iff it has this property:

"All nodes of degree greater than or equal to 3 are surrounded by at most two nodes of degree two or greater."

Finding maximal size subtree of this tree so that it's a special tree.

Input

The first line of the input file contains one integer N — number of nodes in the tree ($0 < N \leq 1000000$). Next N-1 lines contain N-1 edges of that tree — Each line contains a pair (u, v) means there is an edge between node u and node v ($1 \leq u, v \leq N$).

Output

At the first line, output number of nodes in the optimal subtree you found. Next lines, print all edges belong to that subtree, each line contains a pair u v means an edge between node u and node v.

Sample input and output

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

Problem C. The GbAaY Kingdom

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	1-4 seconds
Memory limit:	256 megabytes

Jiajia is the king of the GbAaY Kingdom. He always squeezes his 20 ministers as coolies. There are n cities and m two-way roads connecting cities in the kingdom. Because of the increasing of the oil fee, he want to simplify the road system in the GbAaY Kingdom to save the traffic cost.

Thus, some of roads will be removed. But he requests the ministers guarantee that there is always a path between any two cities. GbAaY Minister Loner suggests Jiajia for the convenience of the traffic management, the farthest distances between cities should be minimal. Unhesitatingly, Jiajia agrees this resolution. As the GbAaY Kingdom's minister (cooly), you must work hard for Jiajia to make the simplification plan.

Input

The first line contains two integers n, m ($1 \leq n \leq 200, n-1 \leq m \leq 20000$). Each line of the following m lines contains three integers u, v, w ($u \neq v$, $0 \leq w \leq 10^5$). They denote there is a road with length w between city u and city v.

Output

The first line contains one integer which is the farthest distance between cities after the simplification. Each line of the follow $n-1$ contains two integers $u, v (u < v)$. They denote there is an road between city u and city v in the simplification plan. If there are many optimal solutions, any of them will be accepted.

Sample input and output

Standard Input	Standard Output
3 3 1 2 1 2 3 1 1 3 1	2 1 2 1 3

Problem D. Let's count 1 2 3

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	0.3-4 seconds
Memory limit:	256 megabytes

Given two integer n, p. 4 kinds of query is needed to solve:
1. Counting the number of labeled unrooted trees with n nodes
2. Counting the number of labeled rooted trees with n nodes
3. Counting the number of unlabeled rooted trees with n nodes
4. Counting the number of unlabeled unrooted trees with n nodes

Calculate the answer modulo p.

Input

Each line contains three integers k, n, p. k denotes which kind of query this case is.
For Kind 1 or Kind 2 query, $1 \leq n \leq 10^9$.
For Kind 3 or Kind 4 query, $1 \leq n \leq 10^3$ and $n \leq p$.
For all queries, $2 \leq p \leq 10^4$ and p is a prime.

Output

For each query, output a line which contains only one integer.

Sample input and output

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

Problem E. Yet another computer network problem

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	10 seconds
Memory limit:	256 megabytes

ACRush and Jelly are practising in the computer room for next TCO. Suddenly, they found the network is very slow. After a few diagnosis, they realized that there are many redundant wires. So they plan to repair the network, change it to an optimal tree topology. And they can't spend too much money to purchase wires. Then.. too easy? Are you thinking about minimum spanning tree?

But the real trouble is the connectors have their own limitation. They can only allow one computer connects with at most B computers.

There are totally 10 cases, arranged in increasing order of the size of N (number of computers). Weight of case i^{th} is $w[i] = i$. We define $infinity = 4 * 10^9$. And in a tree, let's call number of computers that computer i connects with is *degree* of computer i.
For case i^{th} you must show us a satisfied tree with total cost $C[i]$ and maximum degree $M[i]$, considering all computers of that tree. The formula to compute score is as below:
With case i^{th}:
If your $M[i] \leq B$ then *Score[i] = w[i] * C[i]*
If your $M[i] > B$ then *Score[i] = (w[i] + 10) * C[i] * M[i]*

To make the challenge more interesting, with a simple brute force program, we generated 10 upper bound degrees $U[i]$ $(1 \leq i \leq 10)$ for each of 10 cases.
For any case i^{th}:
If your $M[i] > U[i]$ then *Score[i] = infinity*
Finally, $TotalScore = \frac{Score[1]+Score[2]+...+Score[10]}{10}$
Try to minimize the $TotalScore$.

Input

First line contains 3 integers N, M, B – number of computers, number of pairs of computers can be connected and the maximum number of computers that a computer can connect with. $(1 \leq N \leq 10^4$, $1 \leq M \leq 10^5$, $1 \leq B \leq N)$
Next M lines, line i^{th} contains a triple (u_i, v_i, c_i) – means if we want to connect computers u_i and v_i we should purchase a wire, cost c_i $(1 \leq u_i, v_i \leq N$, $1 \leq c_i \leq 20000)$. The wires are bidirectional.

Output

The first line contains 2 numbers — total cost of your tree and the maximum degree in all computers of that tree. Next, print $N-1$ lines, corresponding to $N-1$ edges of the tree, each edge on one line, forms u v.

Sample input and output

Standard Input	Standard Output
3 3 2 1 2 1 2 3 1 1 3 5	2 2 1 2 2 3

Note: You can only submit at most 10 times for this problem.

Problem F. A short vacation in Disneyland

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	0.1-0.5 seconds
Memory limit:	256 megabytes

After a lot of exams at school, Amber and his friends Ahyangyi, Dragon have a short vacation in Hong Kong Disneyland. Many interesting places they want to visit there: resort, castle,... . In each place and between them, there are special bidirectional rails, so that the visitors can drive a small special car, go around and have a sightseeing tour. This rail system is quite optimal it has tree shape ! Each time, you start a new route (or you can call it "path") with a car, you must purchase a new ticket.

Amber and his friends surely want to visit all places, and each place exactly once, so bored to visit one place many times. But the trouble is they don't carry much money. So Amber thinks about a good way to purchase as small number of tickets as possible (i.e. minimal number of routes). We don't care how they can switch cars during their trip.

Now you're given maps of the Disneyland, please help them to find an optimal solution.

Take a look at the figure below:

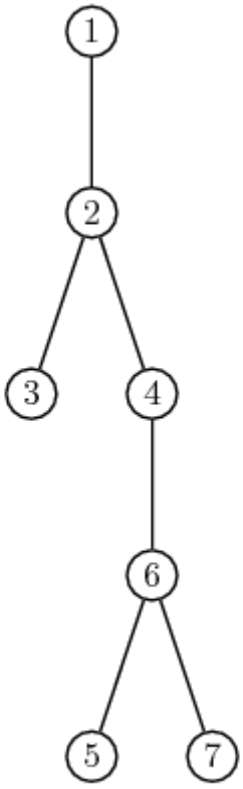


Figure 1: An example map of Disneyland

There are many optimal solutions here and Amber must purchase at least 3 tickets, for 3 disjoint routes. Two possible solutions are:

Solution 1:

1^{st} route: they visit 1 2 3

2^{nd} route: they visit 4

3^{rd} route: they visit 5 6 7
Solution 2:
1^{st} route: they visit 1 2 4 6 5
2^{nd} route: they visit 3
3^{rd} route: they visit 7

Input

There may be many maps in one input file. The first line of file is number of maps T ($0 < T \leq 10$). The following line is blank. Then, there are the descriptions of T maps.
For each map, the first line contains one integer N — number of places in the Disneyland ($0 < N \leq 10000$). We number places from 1 to N. Next $N-1$ lines contain $N-1$ rails between places — Each line contains a pair (u, v) means there is a rail between place u and place v ($1 \leq u, v \leq N$).
There is a blank line after each description.

Output

For each map, the first line, write minimal number of routes K. Next K lines, show out the routes in your solution, each has form $u_1 < blank > u_2 < blank > ...u_m$, means the route starts at place u_1 then visits place u_2,..., ends at place u_m. So between 2 consecutive places in each route must have a rail. If there are many solutions, any of them will be accepted.
There is no blank line after each case.

Sample input and output

Standard Input	Standard Output
1 7 1 2 2 3 2 4 4 6 5 6 6 7	3 1 2 3 4 5 6 7

Problem G. Colorful Lights Party

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	0.1-2 seconds
Memory limit:	256 megabytes

ACRush and his friends want to open a party to celebrate the good result of THU in ICPC 2007. They will use all halls in THU for their party. There are 2 kinds of hall: the small and the large one. In each hall, there is an electronic light system, which forms a tree topology to reduce redundant wires.

1. In a small hall, the light system is a general tree with n lights. The lights are numbered from 1 to n.
2. In a large hall, the light system is obtained from k chains of lights, each chain has length t. The first lights of these k chains are connected with a big light at the central stage of the hall. The big light has id 1, the first light of each chain has id from 2 to $k+1$, then we continue with the second light of each chain, so on...

Take a look at the figure below:

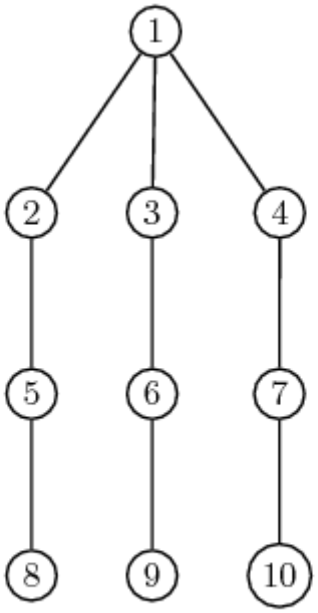


Figure 1: A large hall with 3 chains of lights, each has length 2

ACRush hopes in every hall, each light has an unique color and so do the wires!

For easier to remember and to hang up lights against the walls, he sets a rule:

- For each hall, we number the color from 0 to $n-1$, so each light will get a color id in set {0, 1, ..., n-1}.
- Color id of the wire connects i^{th} light and j^{th} light uniquely identified by the positive difference between color ids of i^{th} light and j^{th} light.

At first view, the rule seems easy, so everyone agrees with him. But it's really tough if the room is quite large, too hard to set colors for lights. After few seconds, ACRush says *"So in this hall, the* 1^{st} *light*

should have color 3, the 2^{nd} *one should have color 0,...".* Well, how can he do it very fast?

How about you ? Let write a program to help ACRush's friends setting colors for lights in all T halls.

Input

The first line of file is T – number of halls in THU ($0 < T \leq 10$). The following line is blank. Then, there are the descriptions of T halls.

For each hall, the first line contains one integer *kind*. *kind* denotes which kind of the current hall: 1 is a small hall, 2 is a large one.
On next lines, there are two cases:

- For Kind 1, first line is n ($1 \leq n \leq 27$) – number of lights. Next n - 1 lines describe wires in this hall. Each line is pair (u, v) – there is a wire between light u and v ($1 \leq u, v \leq n$).
- For Kind 2, only one line contains two numbers k and t ($1 \leq k, t \leq 1000$).

There is a blank line after each description.

Output

For each hall, show us n numbers on one line, i^{th} number is the color id of i^{th} light. If there are many solutions any of them will be accepted. Otherwise, if there is no solution, all color id should be -1. The color ids on one line are separated by exactly one blank, and you'd better not print any redundant blanks.
There is no blank line after each case.

Sample input and output

Standard Input	Standard Output
2 1 3 1 2 2 3 2 2 1	0 2 1 0 4 2 1 3

Problem H. Search in XML

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	0.1 seconds
Memory limit:	256 megabytes

The XML (eXtensible Markup Language) is gaining popularity as a new standard for data representation and exchange on the internet. XML provides a text-based means to describe and apply a tree-based structure to information. The XML document consists of nested elements, some of which usually have attributes and content. But for simplifying this problem, we needn't consider the attributes and content, i.e. only tags allowed. An element typically consists of two tags, a start tag and an end tag. The start tag consists of a name surrounded by angle brackets, like "`<tag>`"; the end tag consists of the same name surrounded by angle brackets, but with a slash preceding the name, like "`</tag>`". The element's content is empty or other sub-element (child) that appears between the start tag and the end tag. Specially, no XML element that has the same tag in its direct sub-elements (children), i.e. All sibling elements have different tag names. The following is an valid example for XML documents.

```
<THU>
    <Team>
        <ACRush></ACRush>
        <Jelly></Jelly>
        <Cooly></Cooly>
    </Team>
    <JiaJia>
        <Team>
            <Ahyangyi></Ahyangyi>
            <Dragon></Dragon>
            <Cooly><Amber></Amber></Cooly>
        </Team>
    </JiaJia>
</THU>
```

For identifying the elements in a document, we number the elements in according to the order that the start tags of the elements appear in the document. For instances, "`THU`" is numbered 1. The first "`Team`" is numbered 2. "`ACRush`" is numbered 3. "`Ahyangyi`" is numbered 8.

The problem of querying XML documents has been given much attention by researchers. Now we are given a querying pattern of XML documents and a text of XML documents. The following is an valid example for pattern.

```
<Team><Cooly></Cooly></Team>
```

And we are requested to find all occurrences of the pattern in the text of XML documents. Here, the pattern occurs at a particular text position if placing the pattern with root element at that text position leads to a situation in which each pattern element overlaps some text element with the same label. Because the sibling elements have different labels, there is only one way to put the pattern into the text.

Input

There are two parts in the input file. The first part is a valid XML documents with exactly one root element. The second part is a valid XML documents as querying pattern with exactly one root element. Please ignore all whitespaces (unvisiable characters) in the input file, i.e. only consider the uppercase and lowercase letter and '`/`', '`<`', '`>`'. Assume XML documents is always strictly a rooted tree. The input file is less than 100kb.

Output

Output all the occurrences of pattern in a text of XML documents. The first line consists of an integer N that denotes the number of the occurrences. Then the next N line, each line consists of an id number of an element that occurs the query pattern.

Please print them in increasing order.

Sample input and output

The input form here looks not very precise, you'd better read the html version.

Standard Input	Standard Output
<THU> <Team> <ACRush></ACRush> <Jelly></Jelly> <Cooly></Cooly> </Team> <JiaJia> <Team> <Ahyangyi></Ahyangyi> <Dragon></Dragon> <Cooly><Amber></Amber></Cooly> </Team> </JiaJia> </THU> <Team><Cooly></Cooly></Team>	2 2 7

Note: You can only submit at most 10 times for this problem.

Problem I. The ants in a tree

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	8-15 seconds
Memory limit:	256 megabytes

In Amber's childhood, he usually liked to observe some little things for tickling his little curiosity. He often found it interesting to climb up a tree, sit on a branch and watch the movement of a group of lovely ants on the branches of the tree.

Amber finds there are n ant holes and m ants in the tree now. Because of his careful observation, he knows all ants' behaviors, the i-th ant wanna travel from one hole s_i to another hole t_i at the speed v_i.

During the ant's travel, if two ants arrive at the same position (meet or chase), they will touch their feelers for exchanging the information about food or danger. Even at the moment that the travel starts or finishes, the ant can also touch other's feelers. But after the travel finishes, the ant will enter into the hole and never touch feelers. What amber wonders is to count the times of touchings during the whole travelling process.

Consider there are $n-1$ branches in the tree. Each branch connects the adjacent ant holes and has a particular length. Assume there is always a path that consists of branches between any two ant holes. Assume that no two ants have the same speeds and the touching doesn't cost any time.

Input

Input consists of multiple testcases. The first line contains one integer t ($1 \le t \le 20$). For each testcase, the input format is following.

The first line contains one integer n ($1 \le n \le 10^6$). In next $n-1$ line, the i-th line contains an integer triple (u_i, v_i, w_i) ($1 \le u_i, v_i \le n, u_i \ne v_i, 1 \le w_i \le 10^3$). The triple means there is a branch with the length w_i between node u_i and node v_i.

The next line contains one integer m ($1 \le m \le 10^3$). In next m line, the i-th line contains an integer triple (s_i, t_i, v_i) ($1 \le s_i, t_i \le n, 1 \le v_i \le 10^6$). The triple means that the i-th ant's travel is from s_i to t_i at the speed v_i.

Output

For each testcase, print a line that consists of an integer that means the times of the feeler touchings.

Sample input and output

Standard Input	Standard Output
1 3 1 2 1 2 3 1 3 1 3 1 3 1 1 1 2 3	2

Problem J. Query on a tree III

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a node-labeled rooted tree with n nodes. Define the query (x, k): Find the node whose label is k-th largest in the subtree of the node x. No two nodes have the same labels.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$). The next line contains n integers l_i ($0 \leq l_i \leq 10^9$) which denotes the label of the i-th node.

Each line of the following $n - 1$ lines contains two integers u, v. They denote there is an edge between node u and node v. Node 1 is the root of the tree.

The next line contains one integer m ($1 \leq m \leq 10^4$) which denotes the number of the queries. Each line of the next m contains two integers x, k. ($k \leq$ the total node number in the subtree of x)

Output

For each query (x, k), output the index of the node whose label is the k-th largest in the subtree of the node x.

Sample input and output

Standard Input	Standard Output
5 1 3 5 2 7 1 2 2 3 1 4 3 5 4 2 3 4 1 3 2 3 2	5 4 5 5

Problem K. JiaJia's balloons

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	1-5 seconds
Memory limit:	256 megabytes

In WinD's birthday, JiaJia opens a small contest for all guests: just typing all characters from 'a' to 'z', who fastest is the winner. The award is a very big bunch of balloons. Of course, JiaJia hopes his girlfriend - WinD can win, since he only carries two bunches with him and the award for this contest looks prettier. Unfortunately, life is not always as he expected, the winner is Amber. So JiaJia can only give WinD his second bunch of balloons. After the party ended, WinD said "I prefer that bunch of balloons to this one!! If you can't give me something the same as it, I don't want to see you anymore!". Well, JiaJia really has a brainstorm.

If we consider balloons as nodes, the colorful strings connect them as edges, and color of each balloon is label of each node then we have each bunch of balloons is a rooted, labeled, ordered tree.

We call T a labeled tree if each node is assigned a symbol from a fixed finite alphabet. And T is an ordered tree if a left-to-right order among siblings in T is given.

JiaJia can do these 3 operators to change the shape of a bunch of balloons (T):

- **Relabel** (recolor a balloon) Change the label of a node v in T.
- **Delete** (remove a balloon) Delete a non-root node v in T with parent p, making the children of v become the children of p. The children are inserted in the place of v as a subsequence in the left-to-right order of the children of p.
- **Insert** (add a balloon) The complement of **Delete**. Insert a node v with any label as a child of node p in T, making v be the parent of a consecutive subsequence of the children of p.

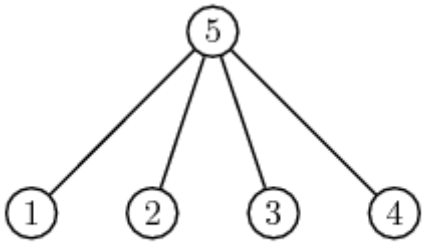


Figure 1: A tree T

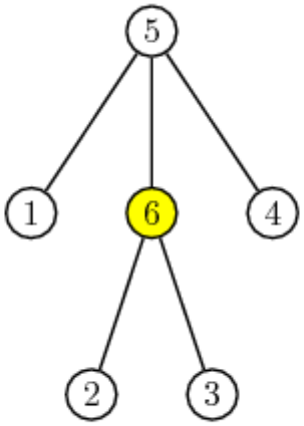


Figure 2: Insert node 6 as a child of 5, and make 6 be the parent of 2, 3

Please help the poor guy JiaJia, use as few number of operators to make WinD's bunch of balloons exactly the same as Amber's bunch of balloons. His girlfriend can't wait any longer. Note that, he can only make changes with WinD's bunch.

Input

The input file contains two bunches of balloons (or trees). The first is of WinD, the second is of Amber.

The first line of the input file contains one integer N - number of balloons in the bunch $T1$ ($1 \leq N \leq 500$). Balloons are numbered from 1 to N. In next N lines, the $i-th$ line contains the ordered children list of the i^{th} balloon. The first integer l_i of the i^{th} line is the color id of the i^{th} balloon ($0 \leq l_i \leq 10^4$). The second integer c_i of the i^{th} line c_i is the number of the children of the i^{th} balloon. Then c_i integers followed, which means the ordered children list of the i^{th} balloon.

The remaining part of the input file is the description of the second bunch of balloons $T2$. The format is the same as $T1$.

Output

Output minimum number of operators JiaJia needs to do.

Sample input and output

Standard Input	Standard Output
3 1 2 2 3 2 0 1 0 2 1 1 2 3 0	2

Explanation We cut the string connected between 1^{st} balloon and 3^{rd} balloons, then change the color of 2^{nd} balloon to 3.