

## Задача А. Декомпозиция

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.6 секунд  
Ограничение по памяти: 256 мегабайт

Рассмотрим дерево  $T$ . Назовем деревом декомпозиции корневое дерево  $D(T)$ . Выберем любую из вершин дерева  $T$ , назовем ее  $r$ . Рассмотрим все компоненты связности дерева  $T$ , после удаления вершины  $r$ :  $S_1, S_2, \dots, S_k$ . Тогда корнем  $D(T)$  будет вершина  $r$ , а детьми  $r$  в  $D(T)$  будут  $D(S_1), D(S_2), \dots, D(S_k)$ .

Вам дано дерево  $T$ . Найдите дерево декомпозиции высоты не более 20. Высота дерева — максимальное число вершин в пути от корня до какой-то вершины.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq 2 \cdot 10^5$ ) — количество вершин дерева.

Следующие  $n - 1$  строк содержат пары чисел  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ), описывающие рёбра дерева.

### Формат выходных данных

Выведите  $n$  чисел, где  $i$ -е — родитель вершины  $i$  в дереве декомпозиции. Если вершина — корень, выведите 0.

### Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3	2 0 2
9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9	0 1 2 2 1 1 6 6 8

## Задача В. Красим дерево

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.6 секунд  
Ограничение по памяти: 256 мегабайт

Дано взвешенное дерево. Вам необходимо выполнять 2 типа запросов:

- «1  $v$   $d$   $c$ » — покрасить все вершины на расстоянии не более  $d$  от  $v$  в цвет  $c$ . Изначально все вершины имеют цвет 0.
- «2  $v$ » — вывести цвет вершины  $v$ .

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в дереве.

Следующие  $n - 1$  содержат тройки чисел  $u_i, v_i, w_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^4$ ).  $i$ -е ребро соединяет вершины  $u_i, v_i$  и имеет вес  $w_i$ .

В следующей строке содержится количество запросов  $q$  ( $1 \leq q \leq 10^5$ ).

Каждая из следующих  $q$  строк содержит запрос какого-то типа:

- 1  $v$   $d$   $c$  ( $1 \leq v \leq n, 0 \leq d \leq 10^9, 0 \leq c \leq 10^9$ ).
- 2  $v$  ( $1 \leq v \leq n$ ).

### Формат выходных данных

Для каждого запроса второго типа выведите ответ на него.

### Примеры

стандартный ввод	стандартный вывод
2	20
1 2 1	10
4	
1 1 1 10	
1 1 0 20	
2 1	
2 2	
5	6
1 2 30	6
1 3 50	0
3 4 70	5
3 5 60	7
8	
1 3 72 6	
2 5	
1 4 60 5	
2 3	
2 2	
1 2 144 7	
2 4	
2 5	

## Задача С. Гоша и праздники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Как известно, жители планеты Иннополис — очень педантичные люди. И даже когда дело касается праздников, они всегда хотят быть уверенными в том, что все пройдёт как по маслу. Так, расписание празднований всех событий на этой планете составлено почти на три миллиона лет вперёд! Гоша — большой любитель праздников. Он решил прилететь в какой-то из городов планеты Иннополис и посетить как можно больше праздников.

На планете Иннополис  $n$  городов, соединённых  $n - 1$  двунаправленными дорогами так, что из любого города планеты можно добраться до любого другого, возможно, посещая другие города. Каждое событие на Иннополисе характеризуется номером города  $c_i$ , в котором оно будет отпраздновано, и номером дня  $d_i$ , в который его будут праздновать.

Гоша настолько везучий человек, что день его прибытия на планету имеет номер 0 в календаре планеты Иннополис, причём исходно он может прилететь в любой город планеты. Гоша решил узнать, какое максимальное количество праздников он может посетить на этой планете. Для этого он обратился за помощью к вам.

### Формат входных данных

В первой строке входного файла задано одно число  $n$  ( $n \geq 1$ ) — количество городов Иннополиса.

В следующих  $n - 1$  строках заданы описания дорог, каждая дорога задается числами  $a_i$ ,  $b_i$  и  $l_i$  ( $1 \leq a_i, b_i \leq n$ ;  $l_i \geq 1$ ) — номера городов, которые соединяет дорога и число дней, необходимых на ее преодоление.

В следующей строке задано число  $m$  ( $m \geq 1$ ) — число праздников на планете.

В следующих  $m$  строках заданы пары чисел  $c_i$  и  $d_i$  ( $1 \leq c_i \leq n$ ;  $d_i \geq 1$ ) — номер города и номер дня, в который пройдёт  $i$ -й праздник.

Ограничения:  $n \leq 2 \cdot 10^5$ ,  $m \leq 2 \cdot 10^5$ ,  $l_i \leq 10^9$ ,  $d_i \leq 10^9$ .

### Формат выходных данных

В единственной строке выходного файла выведите одно число — максимальное количество праздников, которое может посетить Гоша.

## Примеры

стандартный ввод	стандартный вывод
4 1 2 1 2 3 1 2 4 3 4 1 3 2 4 3 1 4 5	3
11 2 1 2 3 2 5 4 1 5 5 2 4 6 5 1 7 1 2 8 3 4 9 6 2 10 7 2 11 2 2 9 1 67 1 34 11 16 5 97 4 70 2 20 2 61 2 26 2 70	8
10 2 1 1 3 2 4 4 2 4 5 3 2 6 4 5 7 5 4 8 3 1 9 6 2 10 7 5 9 7 34 10 82 2 48 3 66 8 98 2 66 3 3 8 59 5 22	8

## Задача D. Гений Евгенийевич

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Гений Евгенийевич давно мечтал о дереве. Сейчас в моде деревья с  $n$  вершинами, на каждой из которых написано число  $x_i$ . Долгожданный день настал, и Гений Евгенийевич смог себе приобрести такое дерево. Вернувшись из магазина после покупки, он начал считать различные величины на этом дереве. Все ему давалось очень просто, пока он не решил найти путь в этом дереве, значение величины  $l(p(u, v)) \cdot \min(x_u, x_v)$  для которого максимально. Здесь  $p(u, v)$  — путь между вершинами  $u$  и  $v$ ,  $l(p)$  — длина пути  $p$  в ребрах.

Уже который день Гений Евгенийевич не выходит на улицу. Его друзья, естественно, начали волноваться. Они просят Вас помочь Гению Евгенийевичу справиться с задачей.

### Формат входных данных

В первой строке входных данных дано число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в дереве.

Во второй строке даны  $n$  чисел  $x_i$  ( $1 \leq x_i \leq 10^9$ ) — числа написанные на вершинах. На  $i$ -й вершине написано число  $x_i$ .

В следующих  $n - 1$ -й строке идут пары чисел  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ) — рёбра в дереве.

### Формат выходных данных

Выведите максимальное значение величины, описанной в условии.

### Пример

стандартный ввод	стандартный вывод
8 9 2 8 8 7 2 1 2 2 1 3 1 4 1 5 2 6 4 7 5 8 5	21

## Задача E. Найти ближайшую

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дано дерево из  $n$  вершин, цвет  $i$ -й вершины равен  $a_i$ . Необходимо обработать  $q$  запросов  $(v_i, c_i)$ : найти расстояние от  $v_i$  до ближайшей вершины цвета  $c_i$ . Расстояние между вершинами — минимальное количество рёбер в пути между ними.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq n \leq 10^5$ ).

Следующая строка содержит  $n - 1$  число  $p_1, \dots, p_{n-1}$  ( $0 \leq p_i < i$ ).  $p_i$  — отец вершины  $i$ .

Следующая строка содержит числа  $a_1, \dots, a_n$  ( $0 \leq a_i < n$ ).

Следующая строка содержит число  $q$  ( $1 \leq q \leq 10^5$ ).

Следующие  $q$  строк содержат числа  $v_i, c_i$  ( $0 \leq v_i < n, 0 \leq c_i < n$ ).

### Формат выходных данных

Для каждого запроса выведите расстояние до ближайшей вершины требуемого цвета, или  $-1$ , если такой нет.

### Пример

стандартный ввод	стандартный вывод
5	0 1 2 -1 2 1 2 1 1
0 1 1 3	
1 2 3 2 1	
9	
0 1	
0 2	
0 3	
1 0	
2 1	
2 2	
3 3	
3 1	
4 2	

## Задача F. Почтовая реформа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Флатландии идет пора реформ. Недавно была проведена реформа дорог, так что теперь по дорогам страны из любого города можно добраться в любой другой, причем только одним способом. Также была проведена реформа волшебников, так что в каждом городе остался ровно один волшебник. Теперь же началась реформа почтовой системы.

Недавно образованное почтовое агентство «Экс-Федя» предлагает уникальную услугу — коллективную посылку. Эта услуга позволяет отправлять посылки жителям всех городов на каком-либо пути по цене обычной посылки. Удивительно, но пользоваться такой услугой стали только волшебники Флатландии, которые стали в большом количестве отправлять друг другу магические кактусы. Агентство столкнулось с непредвиденной проблемой: как известно, все волшебники живут в башнях и мало того, что не строят в них лестницы, так еще время от времени меняют их высоту. Поэтому, чтобы доставить посылку волшебнику, который живет в башне высотой  $h$ , курьеру агентства требуется иметь с собой не менее  $h$  метров веревки.

Вам поручено руководить отделом логистики — по имеющимся данным о высотах башен и об их изменениях вам нужно определять минимальную длину веревки, которую нужно выдать курьеру, который доставляет посылки между городами  $i$  и  $j$ .

### Формат входных данных

Первая строка входного файла содержит число  $n$  — количество городов в Флатландии ( $1 \leq n \leq 50\,000$ ). Во второй строке находится  $n$  положительных чисел, не превосходящих  $10^5$  — высоты башен в городах. В следующих  $n - 1$  строках содержится по два числа  $u_i$  и  $v_i$  — описание  $i$ -й дороги,  $1 \leq u_i, v_i \leq n, u_i \neq v_i$ . В следующей строке содержится число  $k$  — количество запросов ( $1 \leq k \leq 100\,000$ ). В следующих  $k$  строках содержатся описания запросов в следующем формате:

- Уведомление от волшебника из города  $i$  о том, что высота его башни стала равна  $h$ , имеет вид  $! i h, 1 \leq i \leq n, 1 \leq h \leq 10^5$ .
- Запрос от курьера о выдаче веревки для доставки посылок во все города на пути от  $i$  до  $j$  включительно имеет вид  $? i j, 1 \leq i, j \leq n$ .

### Формат выходных данных

Для каждого запроса доставки посылок выведите минимальную длину веревки, которую необходимо выдать курьеру.

## Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 3 2 3 5 ? 1 2 ! 1 5 ? 2 3 ! 3 2 ? 1 2	3 3 5
1 100 5 ! 1 1 ? 1 1 ! 1 1000 ? 1 1 ! 1 1	1 1000



## Задача G. В бухгалтерии опять всё перепутали

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Лула и Пула пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Лула получил зарплату за Пулу, а Пула . . .

Пула не хочет получать за Луну и хочет доказать бухгалтерии, что она не права.

Пула работает в крупной компании «MST Inc.», занимающейся информационным сопровождением «Всеберляндской олимпиады школьников по информатике». В компании «MST Inc.» работает  $n$  сотрудников, причём у каждого из них, кроме самой «MST», есть ровно один непосредственный начальник и несколько (возможно ноль) непосредственных подчинённых.

Всеми начальниками сотрудника компании «MST Inc.» называется множество, состоящее из его непосредственного начальника и множества начальников его непосредственного начальника. Известно, что у каждого сотрудника кроме самой «MST», «MST» входит в множество начальников этого сотрудника.

Множеством подчинённых у сотрудника называется множество, состоящее из него самого и множеств подчинённых у всех непосредственных подчинённых данного сотрудника. В частности, все сотрудники входят в множество подчинённых у «MST».

Каждый месяц каждому сотруднику начисляется зарплата, причём немаленькая, ведь иначе ни один сотрудник не согласился бы работать с «MST». Известно, что в нулевой месяц работы организации, каждому сотруднику заплатили по  $c_i$  бурлей. В качестве поощрения сотрудников «MST» придумала следующее правило: В каждый из следующих  $m$  месяцев берётся сотрудник с номером  $a_i$  и берётся число  $s_i$  — сумма зарплат всех сотрудников во множестве его начальников и подчинённых (включая его самого). Если это число оказывалось слишком большим,  $s_i$  берётся по модулю  $10^9 + 7$ . После этого берётся сотрудник с номером  $b_i$ , и к зарплате всех сотрудников, входящих во множество его начальников и подчинённых (включая его самого) прибавляется число  $s_i$ . С учётом этого изменения платится зарплата в  $i$ -й месяц и пересчитывается зарплата в следующие месяцы.

Вернёмся к Пуле. Пула хочет показать бухгалтерии компании «MST Inc.» что она всё перепутала, а для этого ему надо узнать, сколько же ему должны были заплатить в каждый из месяцев с нулевого по  $m$ -й. К сожалению, в гениальной системе поощрения, разработанной «MST», не может разобраться никто. Поэтому эту задачу поручили вам.

### Формат входных данных

В первой строке входных данных даны 2 числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ) — число сотрудников компании «MST Inc.» и последний день, когда выплачивалась зарплата Пуле.

Во второй строке записано  $n - 1$  число.  $i$ -е из них — номер непосредственного начальника сотрудника номер  $i$  ( $i$  принимает значения от 1 до  $n - 1$ ). При этом «MST» имеет номер 0 и не имеет непосредственного начальника. Пула имеет номер  $n - 1$ .

В третьей строке записано  $n$  чисел  $c_i$  ( $1 \leq c_i \leq 10^9$ ) — зарплата  $i$ -го сотрудника в нулевой день.

В каждой из следующих  $m$  строк записано по 2 числа  $a_i$  и  $b_i$  ( $0 \leq a_i, b_i \leq n - 1$ ) — номер человека, на основе которого происходит поощрение и номер человека, к подчинённым и начальникам которого поощрение применяется (более подробно описано в условии).

### Формат выходных данных

В единственной строке выведите  $m + 1$  число — зарплату Пулы в каждый из дней с 0-го по  $m$ -й. Напоминаем, что Пула имеет номер  $n - 1$ . Обратите внимание, что зарплата **не считается** по модулю  $10^9 + 7$ .

## Примеры

стандартный ввод	стандартный вывод
3 3 0 0 1 1 1 0 0 2 1 1 2	1 4 4 28
4 3 0 1 1 0 1 0 0 0 1 1 3 2 3	0 1 6 20

## Замечание

Пояснение к первому примеру:

В первый день к зарплате каждого сотрудника прибавилось 3 бурля и зарплаты стали соответственно 4, 4, 4.

Во второй день к зарплате сотрудников с номерами 0, 1 прибавилось по 8 бурлей и зарплаты стали соответственно 12, 12, 4.

Во третий день к зарплате сотрудников с номерами 0, 2 прибавилось по 24 бурля и зарплаты стали соответственно 36, 12, 28.

## Задача N. Доставка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Летом Костя решил постажироваться в Яндекс.Еде. К сожалению, его профессиональных навыков не хватило на то, чтобы его взяли в команду разработчиков, поэтому он стал развозить заказы.

В Костином городе ровно  $N$  домов, соединенных между собой  $N - 1$  дорогами таким образом, что из каждого дома можно доехать до любого другого дома ровно одним способом. У каждой дороги есть своя длина  $D_i$  километров.

Для доставки Костя будет пользоваться своими электросамокатами. У каждого Костиного электросамоката есть фиксированная емкость аккумулятора  $T_i$  (будем считать, что емкость аккумулятора измеряется в количестве километров, которые можно проехать от одной зарядки). То есть, Костя проедет  $T_i$  километров без подзарядки, его самокат разрядится и не сможет ехать дальше.

Поскольку некоторые заказы нужно везти очень далеко, на всех Костиных самокатах есть солнечные батареи, с помощью которых можно подзарядить самокат. К сожалению, они работают только тогда, когда самокат стоит на месте, но зато они достаточно качественные, чтобы зарядить самокат за минуту. Каждый раз, когда Костя совершает остановку, он ждет, пока самокат зарядится до конца. Кроме вынужденных остановок, Костя останавливается у каждого дома, мимо которого он проезжает (в том числе, у первого и последнего), потому что соблюдает правила дорожного движения.

Косте за день приходит информация  $Q$  заказах на сегодня. Для каждого заказа скажите, какое минимальное время ему придется заряжать самокат.

### Формат входных данных

Первая строка входного файла содержит числа  $N$  и  $Q$  — количество домов и путей соответственно ( $1 \leq N, Q \leq 100\,000$ ). Следующие  $N - 1$  строки содержат описание дорог. Каждая строка содержит три числа:  $U, V, D$ .  $U$  и  $V$  — это номера домов, соединенных дорогой,  $D$  — длина соответствующей дороги ( $1 \leq U, V \leq N, 1 \leq D \leq 20\,000$ )

Следующие  $Q$  строк описывают маршруты. В каждой из них вводится 3 числа:  $S, F, T$  ( $1 \leq S, F \leq N, S \neq F$ ).  $S$  и  $F$  — это номер дома, от которого Костя начнет выполнение заказа, и номер дома, в который заказ надо привезти.  $T$  — это емкость аккумулятора самоката, на котором Костя будет доставлять этот заказ.

### Формат выходных данных

Для каждого из  $Q$  заказов выведите наименьшее время, которое Косте придется провести в остановках.

### Пример

стандартный ввод	стандартный вывод
7 5	7
1 2 1	9
2 3 2	5
2 4 3	5
4 5 4	12
4 6 5	
4 7 6	
3 7 2	
2 6 1	
5 7 3	
1 4 1	
3 7 1	

## Задача I. Спички детям не игрушка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Лена играет со спичками. Естественный вопрос, посещающий любого школьника, играющего со спичками — а можно ли поджечь спичкой дерево?

Скажем, что дерево — это связный граф без циклов, вершины которого пронумерованы целыми числами  $1, 2, \dots, n$ , в каждой вершине которого также записано некоторое целое число  $p_v$ , являющееся приоритетом вершины  $v$ . Все приоритеты различны.

Оказывается, что если поджечь дерево, то оно, как и можно было ожидать, сгорит целиком. Однако процесс этот не быстрый. Сначала у дерева сгорает лист (листом называется вершина, имеющая ровно одного соседа) с минимальным приоритетом, затем сгорает лист с минимальным приоритетом из оставшихся вершин дерева, и так далее. Таким образом, вершины превращаются в листья и сгорают до тех пор, пока от дерева не останется лишь одна вершина, после чего она тоже сгорает.

Лена приготовила дерево из  $n$  вершин и в каждой вершине записала приоритет  $p_v = v$ . Лене с одной стороны интересно посмотреть, как горит дерево, но с другой она понимает, что если дерево поджечь, оно исчезнет насовсем. Лена добрая девочка, и деревья ей жалко, так что она хочет ограничиться выяснением ответов на некоторые вопросы про процесс сгорания дерева в уме. Лена хочет ответить на  $q$  вопросов, каждый из которых относится к одному из трёх следующих видов:

1. «`up v`», присвоить вершине  $v$  приоритет  $1 + \max(p_1, p_2, \dots, p_n)$
2. «`when v`», выяснить, какой по счёту сгорит вершина  $v$ , если дерево поджечь сейчас;
3. «`compare v u`», выяснить, какая из вершин  $v$  и  $u$  сгорит раньше, если дерево поджечь сейчас.

Заметим, что если приоритеты всех вершин сейчас различны, то и после выполнения запроса «`up v`» они тоже останутся различными. Исходно они различны, поэтому в любой момент времени порядок сгорания листьев определён однозначно.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $2 \leq n \leq 200000$ ,  $1 \leq q \leq 200000$ ) — количество вершин дерева и количество вопросов.

В  $i$ -й из следующих  $n - 1$  строк находятся два целых числа  $v_i, u_i$  ( $1 \leq v_i, u_i \leq n$ ), задающие концы  $i$ -го ребра дерева.

Каждая из оставшихся  $q$  строк содержит операцию одного из трёх типов.

1. «`up v`» ( $1 \leq v \leq n$ ) — присвоить новый приоритет вершине  $v$
2. «`when v`» ( $1 \leq v \leq n$ ) — определить момент сгорания вершины  $v$  для текущего дерева
3. «`compare v u`» ( $1 \leq v, u \leq n, v \neq u$ ) — определить, какая из вершин  $v$  и  $u$  сгорит раньше для текущего дерева.

Гарантируется, что среди запросов хотя бы один имеет тип «`when`» или «`compare`».

### Формат выходных данных

Для каждого запроса типа «`when`» нужно вывести одно целое число от 1 до  $n$  — момент времени, когда сгорит вершина  $v$ .

Для запроса типа «`compare`» выведите  $v$  или  $u$ , в зависимости от того, какая вершина сгорит раньше.

## Примеры

стандартный ввод	стандартный вывод
5 7 1 5 1 2 1 3 4 3 when 1 when 2 when 3 when 4 when 5 compare 2 3 compare 3 4	4 1 3 2 5 2 4
5 5 1 5 1 2 1 3 4 3 up 1 compare 2 4 compare 4 3 compare 3 1 compare 1 5	2 4 3 5

## Задача J. Интерактивная вершина

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

### Это интерактивная задача

У Ильдара есть дерево из  $n$  вершин и он показал его вам. Он выбирает одну вершину  $u$  как специальную вершину, но он не сообщает вам ничего о ней!

Вместо этого, вы можете задавать ему вопросы. Для каждого вопроса вы должны выбрать вершину  $x$ , натуральное число  $k$  и  $k$  вершин  $v_1, v_2, \dots, v_k$  и он вам скажет, правда ли, что  $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$ . Здесь,  $\text{dist}(p, q)$  это количество ребер на простом пути между вершинами  $p$  и  $q$  в дереве.

Вы должны угадать специальную вершину за не больше, чем  $4 \lceil \log_2 n \rceil$  вопросов.

Ильдар очень добрый, поэтому он не будет менять специальную вершину между вашими вопросами (другими словами, интерактор неадаптивный).

**Поскольку ограничения большие и flush это тяжелая операция, убедитесь, что вы не делаете операцию flush слишком часто. Рекомендуется делать flush только после вывода каждого вопроса.**

### Протокол взаимодействия

Процесс взаимодействия начинается с того, что на первой строке вводится целое число  $n$ : количество вершин в дереве Ильдара ( $2 \leq n \leq 200\,000$ ).

Каждая из следующих  $n - 1$  строк содержит два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ), означающих ребро между  $u$  и  $v$ . Гарантируется, что данные ребра образуют дерево.

После этого, вы можете задавать вопросы.

Чтобы задать вопрос, выведите одну строку, содержащую “?  $k$ ” ( $1 \leq k \leq n$ ), целое число  $x$  ( $1 \leq x \leq n$ ) и затем  $k$  различных целых чисел  $v_1, v_2, \dots, v_k$  ( $1 \leq v_i \leq n$ ). Разделяйте соседние числа в строке ровно одним пробелом. Затем сделайте flush выходного потока.

После каждого вопроса, считайте одно целое число  $ans \in \{0, 1\}$ . Если  $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$ , тогда  $ans$  будет равно 1. Иначе,  $ans$  будет равно 0.

Когда вы нашли специальную вершину  $u$  ( $1 \leq u \leq n$ ), выведите одно целое число “!  $u$ ”, сделайте flush выходного потока и завершите работу программы.

Ваше решение получит Wrong Answer или Time Limit Exceeded если вы сделаете больше чем  $4 \lceil \log_2 n \rceil$  вопросов.

Ваше решение получит Idleness Limit Exceeded если оно не совершает никаких действий или не делает flush выходного потока.

Чтобы сделать flush выходного потока, вы можете использовать:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- используйте документацию других языков.

## Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 1 2 1 3 1 4 1 5 1	? 4 1 2 3 4 5 ! 1
5 1 2 1 3 1 4 1 5 0 0 0 0	? 4 1 2 3 4 5 ? 3 1 2 3 4 ? 2 1 2 3 ? 1 1 2 ! 2
7 1 2 2 3 3 4 4 5 3 6 6 7 1	? 3 3 5 7 1 ! 3

## Замечание

