Задача A. Выбор вершин дерева

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Дан граф, являющийся деревом. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества посчитаем количество вершин в нём. Каково максимальное из этих количеств?

Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i, не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер i—p_i для всех вершин i, кроме корня.

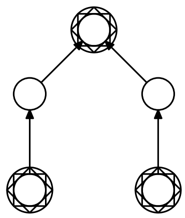
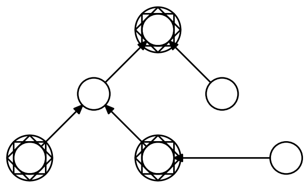
В первой строке записано целое число n — количество вершин в графе ($1 \leqslant n \leqslant 100$). В следующих n строках задан граф. В i-й из этих строк записано целое число p_i — номер вершины-предка i-й вершины. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leqslant p_i \leqslant n$.

Гарантируется, что заданный граф является деревом.

Формат выходных данных

В первой строке выведите одно число — максимальное количество вершин в допустимом множестве.

Примеры

стандартный ввод	**стандартный вывод**	**Пояснение**
5 0 1 1 2 3	3	
6 5 6 5 1 0 1	3	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальным количеством вершин в нём.

Задача B. Выбор вершин взвешенного дерева

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i, не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер $i - p_i$ для всех вершин i, кроме корня.

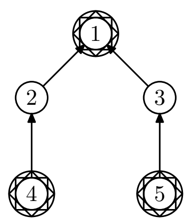
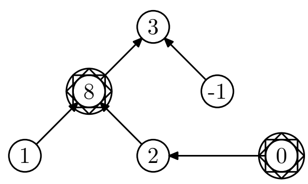
В первой строке записано целое число n — количество вершин в графе ($1 \leqslant n \leqslant 100$). В следующих n строках задан граф. В i-й из этих строк записаны через пробел два целых числа p_i и q_i; здесь p_i — номер вершины-предка i-ой вершины, а q_i — число, записанное в этой вершине. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leqslant p_i \leqslant n$. Числа q_i не превосходят $10\,000$ по абсолютной величине.

Гарантируется, что заданный граф является деревом.

Формат выходных данных

В первой строке выведите одно число — максимальную сумму чисел в допустимом множестве.

Примеры

стандартный ввод	стандартный вывод	Пояснение
5 0 1 1 2 1 3 2 4 3 5	10	
6 5 8 6 0 5 -1 1 1 0 3 1 2	8	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальной суммой чисел в вершинах.

Задача C. Ограниченный выбор вершин взвешенного дерева

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Дан ориентированный граф, являющийся корневым деревом. Дуги графа направлены от сыновей к отцам, и из каждой вершины можно по дугам прийти в корень. В вершинах графа написаны целые числа.

Множество вершин графа называется *недопустимым*, если оно содержит **более** k сыновей какой-либо вершины.

Множество вершин графа называется *допустимым*, если оно не является недопустимым, а кроме того, никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

Формат входных данных

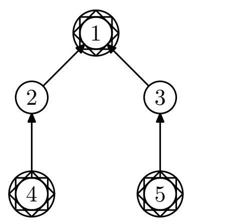
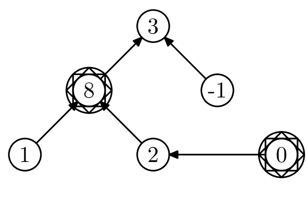
В первой строке записаны через пробел два целых числа n и k — количество вершин в графе и ограничение на количество взятых сыновей какой-либо вершины, соответственно ($1 \leqslant n \leqslant 100$, $0 \leqslant k \leqslant n$). В следующих n строках задан граф. В i-й из этих строк записаны через пробел два целых числа p_i и q_i; здесь p_i — номер вершины-предка i-ой вершины, а q_i — число, записанное в этой вершине. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leqslant p_i \leqslant n$. Числа q_i не превосходят $10\,000$ по абсолютной величине.

Гарантируется, что заданный граф является корневым деревом.

Формат выходных данных

В первой строке выведите одно число — максимальную сумму чисел в допустимом множестве.

Примеры

стандартный ввод	*стандартный вывод*	**Пояснение**
5 2 0 1 1 2 1 3 2 4 3 5	10	
6 1 5 8 6 0 5 -1 1 1 0 3 1 2	8	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальной суммой чисел в вершинах.

Задача D. Выбор k вершин взвешенного дерева

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа, состоящие ровно из k вершин. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i, не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер $i - p_i$ для всех вершин i, кроме корня.

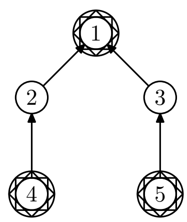
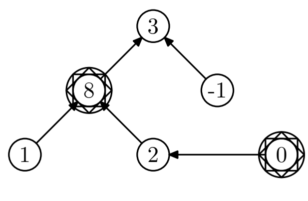
В первой строке записаны через пробел два целых числа n и k — количество вершин в графе и размер множеств, соответственно ($1 \leqslant n \leqslant 100$, $0 \leqslant k \leqslant n$). В следующих n строках задан граф. В i-й из этих строк записаны через пробел два целых числа p_i и q_i; здесь p_i — номер вершины-предка i-ой вершины, а q_i — число, записанное в этой вершине. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leqslant p_i \leqslant n$. Числа q_i не превосходят $10\,000$ по абсолютной величине.

Гарантируется, что заданный граф является деревом, а также что в нём существует хотя бы одно допустимое множество, состоящее ровно из k вершин.

Формат выходных данных

В первой строке выведите одно число — максимальную сумму чисел в допустимом множестве из k вершин.

Примеры

стандартный ввод	**стандартный вывод**	**Пояснение**
5 3 0 1 1 2 1 3 2 4 3 5	10	
6 2 5 8 6 0 5 -1 1 1 0 3 1 2	8	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество из k вершин с максимальной суммой чисел в них.

Задача E. Раскраска дерева

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Дано корневое дерево из n вершин. Корень дерева имеет ровно три сына. Все вершины, за исключением корня и листьев, имеют ровно по два сына.

Все листья дерева покрашены в один из трёх цветов. Необходимо раскрасить остальные вершины дерева в эти же три цвета так, чтобы смежные вершины были окрашены в различные цвета, либо определить, что такой раскраски не существует.

Формат входных данных

Входные данные состоят из одного или нескольких тестовых случаев. Каждый тестовый случай задаётся следующим образом.

Сначала в отдельной строке записано целое число n ($4 \leqslant n \leqslant 300\,000$). В следующих $n-1$ строках описывается дерево. В i-й из этих строк задано одно целое число p ($1 \leqslant p \leqslant i$) — номер отца вершины с номером $i+1$. Корень дерева всегда имеет номер 1, а остальные вершины пронумерованы от 2 до n.

Далее в отдельной строке дано целое число m — количество листьев данного дерева. В следующих m строках перечислены листья дерева, по одному в строке. Описание каждого листа состоит из его номера (целое число от 2 до n) и цвета (целое число от 1 до 3).

Гарантируется, что сумма всех n во входных данных также не превосходит $300\,000$. После описания каждого тестового случая следует пустая строка. Входные данные завершаются строкой, на которой вместо n записано число 0.

Формат выходных данных

В ответ на каждый тестовый случай выведите в отдельной строке «YES», если раскраска существует, и «NO» в противном случае. Если раскраска существует, то в следующей строке выведите n чисел. В этой строке i-е число должно задавать цвет вершины с номером i. Если раскрасок существует несколько, то можно вывести любую из них.

Примеры

стандартный ввод	**стандартный вывод**
4 1 1 1 3 2 3 3 1 4 2 0	NO

стандартный ввод	**стандартный вывод**
6 1 1 1 2 2 4 5 2 6 3 3 1 4 1 0	YES 2 1 1 1 2 3

Задача F. Красно-чёрное дерево

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Рассмотрим двоичное корневое дерево: в этом дереве выделена вершина-корень, все рёбра ведут в направлении от корня, и у каждой вершины от нуля до двух детей. Будем считать, что из каждой вершины выходят ровно два ребра: если при этом у неё меньше двух детей, оставшиеся рёбра ведут в пустоту.

Будем называть дерево *красно-чёрным*, если выполнены следующие условия:

- Каждая вершина покрашена либо в красный, либо в чёрный цвет.
- В дереве нет ребра с двумя красными концами. Пустота считается чёрной.
- Рассмотрим все пути по рёбрам, идущие из корня в пустоту. Количество чёрных вершин на всех таких путях одинаково.

Аналогичную раскраску можно использовать в двоичных деревьях поиска для их балансировки.

Дано непустое двоичное корневое дерево. Покрасьте его так, чтобы оно стало красно-чёрным, или выясните, что это невозможно.

Формат входных данных

В первой строке задано целое число n ($1 \leqslant n \leqslant 500$) — количество вершин в дереве. Вершины пронумерованы числами от 1 до n.

В следующей строке записаны через пробел n целых чисел: $p_1, p_2, \ldots, p_n$ ($0 \leqslant p_i \leqslant n$). Число $p_i > 0$ означает, что вершина i — ребёнок вершины p_i. Если же $p_i = 0$, то i — корень дерева.

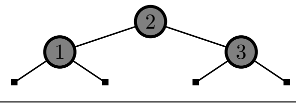
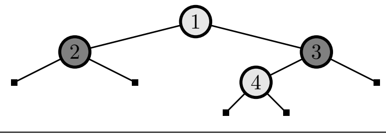
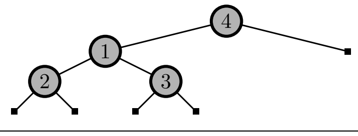
Гарантируется, что во входных данных корректно задано двоичное корневое дерево: корень ровно один, у каждой вершины от нуля до двух детей, а кроме того, из корня можно, двигаясь по рёбрам, попасть во все остальные вершины.

Формат выходных данных

Если можно покрасить заданное дерево так, чтобы оно стало красно-чёрным, выведите любую такую раскраску в виде строки из n символов. Символ на i-й позиции должен быть равен «R», если вершина i красная, и «B», если она чёрная.

Если же покрасить дерево невозможно, выведите слово «Impossible».

Примеры

стандартный ввод	**стандартный вывод**	**Пояснение**
3 2 0 2	BBB	
4 0 1 1 3	RBBR	
4 4 1 1 0	Impossible	

Задача G. Длина подпоследовательности

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вам требуется написать программу, которая по заданной последовательности находит длину её максимальной невозрастающей подпоследовательности (то есть такой последовательности чисел $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ ($i_1 < i_2 < \ldots < i_k$), что $a_{i_1} \geqslant a_{i_2} \geqslant \ldots \geqslant a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leqslant n \leqslant \mathbf{239\,017}$). В последующих строках идут сами числа последовательности a_i, отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать число k — длину максимальной невозрастающей подпоследовательности.

Пример

стандартный ввод	**стандартный вывод**
5 5 8 10 4 1	3

Задача H. Невозрастающая подпоследовательность

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (то есть такую последовательность чисел $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ ($i_1 < i_2 < \ldots < i_k$), что $a_{i_1} \geqslant a_{i_2} \geqslant \ldots \geqslant a_{i_k}$ и не существует последовательности с теми же свойствами длины $k+1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leqslant n \leqslant \mathbf{239\,017}$). В последующих строках идут сами числа последовательности a_i, отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке число k — длину максимальной невозрастающей подпоследовательности. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности i_j, образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Пример

стандартный ввод	***стандартный вывод***
5 5 8 10 4 1	3 2 4 5

Задача I. Новая модель телефона

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Компания Gnusmas разработала новую модель мобильного телефона. Основное достоинство этой модели — ударопрочность: её корпус сделан из особого сплава, и телефон должен выдерживать падение с большой высоты.

Компания Gnusmas арендовала n-этажное здание и наняла экспертов, чтобы те при помощи серии экспериментов выяснили, с какой высоты бросать телефон можно, а с какой — нельзя. Один эксперимент заключается в том, чтобы бросить телефон с какого-то этажа и посмотреть, сломается он от этого или нет. Известно, что любой телефон этой модели ломается, если его сбросить с x-го этажа или выше, где x — некоторое целое число от 1 до n, включительно. Задача экспертов заключается в том, чтобы узнать число x и передать его рекламному отделу компании.

Задача осложняется тем, что экспертам предоставлено всего k образцов новой модели телефона. Каждый телефон можно бросать сколько угодно раз, пока он не сломается; после этого использовать его для экспериментов больше не удастся.

Подумав, эксперты решили действовать так, чтобы минимизировать максимально возможное количество экспериментов, которое может потребоваться произвести. Чему равно это количество?

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа n и k — количество этажей в здании и количество образцов новой модели телефона ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant k < n$).

Формат выходных данных

В выходной файл выведите единственное число — минимальное количество экспериментов, которое потребуется совершить, чтобы узнать число x и использовать не более k телефонов. Если решить задачу невозможно, выведите вместо этого -1.

Примеры

стандартный ввод	**стандартный вывод**
4 2	2
4 1	3

Пояснения к примерам

В первом примере сначала следует бросить телефон со второго этажа. Если он сломается, то второй бросок следует сделать с первого этажа. В случае поломки станет известно, что $x = 1$. Иначе мы узнаем, что $x = 2$.

Если же при броске со второго этажа телефон не сломался, бросим телефон с третьего этажа. При поломке будет ясно, что $x = 3$. Иначе из условия $1 \leqslant x \leqslant 4$ следует, что $x = 4$.

Всего будет сделано два эксперимента. В них будет использовано не более чем два телефона.

Во втором примере следует сначала бросить единственный данный нам телефон с первого этажа, если он не сломается, то со второго, а если опять не сломается, то с третьего. При первой же поломке мы узнаем точное значение x. Если после трёх бросков телефон так и не сломался, то $x = 4$.

Задача J. Аттракцион

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

В парке поставили экспериментальную модель аттракциона. Посетители рассаживаются по местам, затем аттракцион запускают, и посетители с головокружительной скоростью носятся по причудливым траекториям. К сожалению, аттракцион не всегда работает: иногда он ломается, и все, кто в этот момент был на аттракционе, уходят недовольными и больше в парк не приходят. Было установлено, что аттракцион ломается тогда и только тогда, когда на нём катается не меньше, чем x человек, для некоторого неизвестного заранее, но фиксированного целого числа x ($1 \leqslant x \leqslant n$, где n — общее число мест на аттракционе).

Хозяин парка аттракционов — циничный человек, и он не очень расстроится, если несколько посетителей будут недовольны поломками на аттракционе. Гораздо больше его заботит расход электроэнергии, требуемой для запуска аттракциона. Количество энергии, требуемое для однократного запуска аттракциона, постоянно и не зависит от количества занятых мест и от того, случится ли при работе аттракциона поломка.

Хозяин парка хочет сделать как можно меньше запусков аттракциона, прежде чем ему станет известно, чему в точности равно число x. В его власти количество посетителей при каждом новом запуске аттракциона. Однако хозяин парка знает, что, если в ходе его экспериментов недовольных окажется больше, чем k человек, они соберутся вместе и напишут жалобу в министерство. Конечно, для хозяина парка такой поворот событий совершенно неприемлем.

Примерьте на себя шкуру холодного расчётливого циника и посчитайте, какое минимальное количество запусков потребуется для решения поставленной задачи.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа n и k — количество мест на аттракционе и максимальное возможное количество недовольных ($1 \leqslant n \leqslant 600$, $0 \leqslant k \leqslant 10^9$).

Формат выходных данных

В выходной файл выведите единственное число — минимальное количество запусков аттракциона, которое потребуется совершить, чтобы узнать число x и сделать не более k человек недовольными. Если решить задачу невозможно, выведите вместо этого -1.

Примеры

стандартный ввод	***стандартный вывод***
`4 3`	`2`
`4 2`	`-1`

Пояснения к примерам

В первом примере сначала следует запустить аттракцион с двумя посетителями. Если он сломается, во второй запуск следует посадить на аттракцион одного человека. В случае поломки всего недовольных будет трое и станет известно, что $x = 1$. В случае успешного запуска недовольных будет двое и мы узнаем, что $x = 2$.

Если же с двумя посетителями аттракцион не сломался, запустим его для трёх человек. При поломке будет три недовольных и $x = 3$. При успешном запуске недовольных нет и из условия $1 \leqslant x \leqslant 4$ получаем, что $x = 4$.

Определить x быстрее, чем за два запуска, в первом примере не удастся.

Во втором примере невозможно посадить на аттракцион более двух человек, из-за чего оказывается, что случаи $x = 3$ и $x = 4$ неразличимы.