

## Задача А. Битва с боссами

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

Зельда опять бродит по подземельям, и ей очень хочется победить всех боссов, которые в них прячутся. Однако не всех боссов Зельда способна победить сразу, поэтому она тщательно выбирает, с кем ей сразиться сначала.

Иерархия боссов представляет собой подвешенное дерево, где вершины — это боссы, а рёбра — отношения начальник/подчиненный между боссами. Если в дереве вершина  $v$  является ребенком вершины  $u$ , то босс с номером  $u$  — непосредственный начальник босса с номером  $v$ . Будем называть непосредственного начальника *лордом*, а непосредственного подчиненного — *служгой*.

Зельда придерживается следующей стратегии:

1. Она выбирает таких  $u$  и  $v$ , что  $v$  — **единственный** слуга босса  $u$ .
2. Затем она расправляется с боссом  $v$ .
3. После этого все слуги босса  $v$  переходят в подчинение  $u$ .
4. Процесс повторяется.

Однако боссам не нравится, что после такого действия некоторым из начальников приходится перерабатывать, так как появляется слишком много слуг. Для того чтобы оптимизировать нагрузку, структура боссов перестраивается:

1. Выбирается босс  $w$  с самым большим числом слуг (если таких несколько, то имеющий максимальный номер из них).
2. Выбирается его слуга  $y$  с наибольшим номером.
3.  $y$  становится лордом всех остальных слуг  $w$ .
4.  $y$  при этом остается слугой  $w$ .

После этого Зельда снова ищет боссов, у которых ровно один подчиненный, и побеждает их, затем иерархия боссов снова перестраивается, и так до тех пор, пока Зельда может найти, кого ей победить. Помогите Зельде понять, сколько боссов останется после всех ее побед.

### Формат входных данных

В первой строке дано целое число  $n$  — число боссов ( $1 \leq n \leq 10^5$ ).

Далее дано  $n$  строк, описывающих слуг боссов. Строка с номером  $i$  состоит из числа  $k_i$  (числа слуг босса номер  $i$ ) и следующих за ним  $k_i$  различных целых чисел  $s_1, \dots, s_{k_i}$  — номеров этих слуг ( $1 \leq s_j \leq s_n, s_j \neq i$ ).

Гарантируется, что описанная структура задает подвешенное дерево.

### Формат выходных данных

В первой строке выведите  $m$  — число боссов, которые не будут побеждены Зельдой. Во второй строке через пробел выведите  $m$  чисел — номера боссов, которых Зельде не удастся победить.

### Пример

стандартный ввод	стандартный вывод
4	1
2 2 3	1
0	
1 4	
0	

## Задача В. Оптимизация заклинаний

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

Принцесса Зельда готовится сражаться с Нуллом и оптимизирует работу Жезла Трифа: она настроила жезл так, что теперь он может вызывать пару эхо одним заклинанием.

У каждого эхо есть имя, а каждое заклинание — это определенное волшебное слово. Известно, что эхо довольно придирчивы и реагируют только на заклинания, являющиеся префиксами их имен.

Всего у Зельды есть  $n$  эхо,  $i$ -е из которых имеет имя  $s_i$ , и  $m$  волшебных слов заклинаний,  $i$ -е из которых равно  $t_i$ . Формально правила следующие: пару эхо  $i$  и  $j$  можно вызвать заклинанием номер  $k$ , если

- $t_k$  является префиксом  $s_i$ ;
- $t_k$  является префиксом  $s_j$ ;
- не существует строки длиннее, чем  $t_k$ , которая также является и префиксом  $s_i$ , и префиксом  $s_j$ .

Теперь Зельде интересно проверить работу жезла перед финальной битвой и убедиться, что доступные ей волшебные слова дают ей достаточную гибкость в битве. Определите, для скольки пар эхо Зельда имеет заклинание, способное эту пару эхо вызвать.

### Формат входных данных

В первой строке ввода даны два целых числа  $n$  и  $m$  — число эхо и заклинаний соответственно ( $1 \leq n, m \leq 2 \cdot 10^5$ ).

Во  $i$ -й из следующих  $n$  строк дано имя  $i$ -го эхо — строка из маленьких латинских букв (от 'a' до 'z').

Во  $j$ -й из следующих  $m$  строк дано  $j$ -е заклинание — также строка из маленьких латинских букв.

Суммарная длина строк из одного набора не превосходит  $10^6$ .

### Пример

стандартный ввод	стандартный вывод
3 4 abacaba aba bbbb a aba b abacaba	1

## Задача С. Застывший Мир

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

В глубинах Застывшего Мира, где неподвижно даже время, скрывается Линк. Силы тьмы похитили его, и только Зельда, храбрая принцесса Хайрула, может спасти его.

Зельда обладает древним артефактом — Жезлом Трифа, который при активации загорается огнем, если герой Линк находится на расстоянии не больше  $2d$  от жезла. Правда, Зельда читала инструкцию по использованию Жезла очень давно и уже забыла, чему равно  $d$ , она только помнит, что  $d$  целое и лежит в отрезке от 1 до  $10^5$ .

Сейчас Зельда находится в точке  $(0, 0)$  и знает, что Линк находится на расстоянии не больше  $d$  от нее, то есть если активировать жезл, то он загорится.

Линк спрятан силами тьмы в некоторой точке  $(x_0, y_0)$  с необязательно целыми координатами, причем  $-10^5 \leq x_0, y_0 \leq 10^5$ . Зельда хочет как можно быстрее найти его. Для этого она может не более 400 раз перейти в произвольную точку  $(x, y)$  ( $-10^6 \leq x, y \leq 10^6$ ) координатной плоскости и активировать Жезл, чтобы проверить, находится ли Линк на расстоянии не больше  $2d$ .

Помогите Зельде найти координаты Линка с абсолютной погрешностью каждой из координат точки не более  $10^{-3}$ .

### Протокол взаимодействия

Это интерактивная задача.

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Каждый запрос представляется в виде «?  $x$   $y$ », что означает перейти в точку с вещественными координатами  $x$  и  $y$  ( $-10^6 \leq x, y \leq 10^6$ ) и активировать Жезл Трифа. В качестве ответа на запрос вы получите одну из двух строк:

- «YES», если Линк находится на расстоянии не больше  $2d$  от точки с координатами  $(x, y)$ ;
- «NO» в противном случае.

Чтобы вывести ответ на задачу, напечатайте «!  $x_0$   $y_0$ », где  $x_0$  и  $y_0$  — предполагаемые координаты места, где спрятан Линк. Этот вывод не учитывается в количестве запросов.

Если в какой-то момент ваша программа превышает лимит в 400 запросов или вы выводите запрос в точке за пределами доступной области, интерактор завершится с вердиктом WA (Wrong Answer).

Не забывайте после каждого запроса выводить символ перевода строки (`'\n'`) и сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт TL (Time Limit Exceeded) или IL (Idleness Limit Exceeded).

## Пример

стандартный ввод	стандартный вывод
? 1 0	YES
? 3 0	YES
? -1 1	NO
? 1 1	YES
? 0 2	NO
? 2 1	YES
? 2 1.000001	NO
? 0 -1.000001	NO
! 1 0	

## Замечание

В первом примере из условия одно из возможных значений  $d$  равно 1. Тогда если мы находимся на расстоянии не больше 2 от  $(1, 0)$ , мы получаем ответ YES, а иначе NO.

## Задача D. Простая игра

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

После долгих путешествий и приключений по Хайрулу Линк и Зельда решили отдохнуть и отвлечься от боев с монстрами и спасения королевства. Линк предложил Зельде сыграть в интересную числовую игру, которую он сам недавно узнал.

Правила игры довольно просты: есть два целых положительных числа  $A$  и  $B$  и число  $k$ . Игроки ходят по очереди, начинает Зельда. За ход игрок может сделать ровно одно из двух действий:

- Вычесть целое число от 1 до  $k$  из числа  $A$ ;
- Прибавить целое число от 1 до  $k$  к числу  $B$ .

При этом, в процессе игры числа  $A$  и  $B$  должны оставаться положительными; если же игрок сделал такой ход, что одно из чисел перестало быть положительным, он проигрывает. Выигрывает же в игре тот, после чьего хода сумма чисел  $A$  и  $B$  стала простым числом.

Зельда хочет показать, что не уступает Линку в остроумии. Подскажите ей, может ли она выиграть в игре вне зависимости от игры соперника?

### Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка ввода содержит одно целое число  $t$  — количество наборов входных данных ( $1 \leq t \leq 10$ ).

В следующих  $t$  строках описываются наборы входных данных: каждый из них состоит из одной строки, содержащей три числа  $A, B, k$  — числа  $A$  и  $B$  в начале игры и число  $k$  ( $1 \leq A, B, k \leq 10^9$ ).

Гарантируется, что  $A + B$  изначально не является простым числом.

### Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке «Yes», если Зельда сможет выиграть с такими начальными данными, и «No» иначе.

### Пример

стандартный ввод	стандартный вывод
4	No
1 7 2	Yes
3 5 2	No
3 6 1	Yes
253309356 182963670 154154154	

## Задача Е. Чудеса природы

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

Поверхность мира Хайрул усеяна холмами и ущельями, полными опасностей.

Зельда ориентируется по карте Хайрула, на которой отмечено  $n$  локаций. Каждая локация имеет свою высоту над уровнем моря  $a_i$ .

*Холм* — это подпоследовательность локаций (не обязательно идущих подряд, но имеющих возрастающие номера), высоты которых сначала строго возрастают, а затем строго убывают. Иными словами, последовательность  $i_1, \dots, i_k$  называется холмом, если  $i_1 < i_2 < \dots < i_k$  и есть некоторое  $j$  от 1 до  $k$ , что  $a_{i_1} < a_{i_2} < \dots < a_{i_j}$  и  $a_{i_j} > a_{i_{j+1}} > \dots > a_{i_k}$ . Обратите внимание, что холм может состоять даже из одной локации.

Чтобы спасти Линка и все королевство от неизвестного врага, порождающего порталы в Застывший Мир, необходимо подсчитать количество всех возможных холмов в Хайруле. Казалось бы, это не сильно связанные вещи, но Зельда знает, что делает.

Поскольку холмов может быть очень много, Зельде достаточно найти остаток от деления количества холмов на  $10^9 + 7$ . Помогите принцессе справиться с этой задачей.

### Формат входных данных

В первой строке ввода дано целое положительное число  $n$  — число локаций в мире Хайрул ( $1 \leq n \leq 10^5$ ).

Во второй строке перечислены  $n$  чисел  $a_i$  — высоты локаций ( $0 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число — остаток от деления общего числа холмов на  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
5 3 2 5 1 4	20
3 2 1 3	6

## Задача F. Закрывать порталы

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

В Хайруле продолжают открываться порталы в Застывший Мир, и вот, пока Зельда сидела в тюрьме, три портала разрослись до невероятных размеров. Представьте Хайрул как множество из  $n$  городов, связанных  $m$  двухсторонними дорогами так, что из любого города можно добраться до любого другого. Тогда  $i$ -й портал поглотил уже  $k_i$  городов  $a_{i,1}, a_{i,2}, \dots, a_{i,k}$ .

Известно, что города, поглощенные каждым порталом, образуют связанные множества. То есть если оставить в Хайруле только города, поглощенные каким-то конкретным порталом, и дороги между ними, то между любыми двумя из них будет существовать путь. Также известно, что никакой город не накрыт более чем одним порталом, то есть  $a_{i,p} \neq a_{j,q}$  при  $i \neq j$ .

Для того, чтобы закрыть порталы и спасти жителей, Зельде достаточно соединить их так, чтобы порталы дестабилизировались и закрылись сами по себе, вернув все на свои места. Для этого ей необходимо пометить какие-то города своим Жезлом Трифа, чтобы множество этих городов вместе с городами, поглощенными порталами, образовывало связное множество.

Иными словами, необходимо выбрать множество городов  $B$  так, чтобы любые два города из множества  $S = B \cup \{a_{i,j}\}$  были соединены путем, проходящим только по городам из  $S$ .

Разумеется, самый очевидный способ — это пометить все города в Хайруле, но Зельду такое решение не устроит. Найдите подходящее множество  $B$  минимального размера.

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $m$  — количество городов и дорог в Хайруле ( $1 \leq n, m \leq 2 \cdot 10^5$ ).

Следующие  $b$  строк описывают порталы: по две строки на каждый. Первая строка описания  $i$ -го портала содержит число  $k_i$  — количество городов, которые им покрыты ( $1 \leq k_i \leq n - 2$ ). Во второй строке перечислены  $k_i$  различных целых чисел  $a_{i,j}$  — номера городов, поглощенных  $i$ -м порталом ( $1 \leq a_{i,j} \leq n$ ).

Гарантируется, что каждое число от 1 до  $n$  встречается среди  $a_{i,j}$  не более одного раза.

В  $i$ -й из следующих  $m$  строк даны два целых числа  $u_i$  и  $v_i$  — номера городов, соединенных  $i$ -й дорогой ( $1 \leq u_i, v_i \leq n$ ).

Гарантируется, что от любого города можно добраться до любого другого по данным дорогам.

### Формат выходных данных

В первой строке выведите целое число  $b$  — минимальное число городов, которые Зельде придется пометить Жезлом Трифа. Если порталы уже связаны и пометить города не надо, выведите 0.

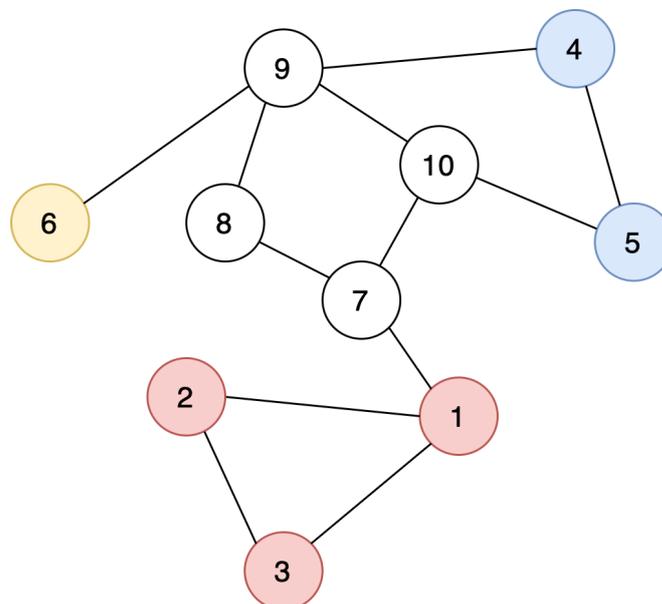
Во второй строке выведите  $b$  различных целых чисел от 1 до  $n$  — номера этих городов. Никакой поглощенный порталом город не должен быть помечен.

## Примеры

стандартный ввод	стандартный вывод
4 3 1 1 1 2 1 3 4 1 4 2 4 3	1 4
10 12 3 1 2 3 2 4 5 1 6 1 2 2 3 1 3 4 5 1 7 7 8 8 9 9 6 9 10 7 10 10 5 4 9	3 9 7 10

## Замечание

Иллюстрация для второго примера из условия приведена на рисунке ниже. В качестве ответа подойдут города 7, 8, 9 или 7, 9, 10.



## Задача G. Лечебный смузи

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

Зельда нашла Линка (ура!), но теперь ей нужно сварить лечебный смузи, чтобы нейтрализовать влияние черной магии и поправить здоровье друга.

Приготовление смузи — это огромный квест, перед началом которого Зельда выбирает  $n$  ингредиентов, заданных числами от 0 до  $10^9$ , и упорядочивает их в последовательность  $a_i$ . Разрешается брать повторяющиеся ингредиенты, то есть  $a_i = a_j$ .

Затем принцессе необходимо найти секретный ингредиент, номер которого вычисляется следующим образом:

- Для каждого  $i$  от 1 до  $n$  для последовательности  $a_1, \dots, a_{i-1}, a_i$  определяется  $b_i$  — минимальный номер ингредиента, которого в этой последовательности нет (обозначается как  $\text{mex}(a_1, \dots, a_i)$ ). Например,  $\text{mex}(1, 7, 2, 2, 0, 5, 4) = 3$  и  $\text{mex}(1, 4, 5, 2, 3) = 0$ .
- От полученной последовательности  $b_1, \dots, b_n$  также вычисляется  $\text{mex}$ .
- И полученное в итоге значение  $\text{mex}(b_1, \dots, b_n)$  берется в качестве номера секретного ингредиента.

Зельда уже выбрала набор из  $n$  ингредиентов, но заметила, что от их порядка может меняться номер секретного ингредиента. Известно, что чем больше его номер, тем более высокими целительными свойствами будет обладать смузи, приготовленный с его использованием.

Помогите принцессе упорядочить выбранные ингредиенты так, чтобы соответствующий полученной последовательности ингредиентов номер секретного ингредиента был как можно больше.

### Формат входных данных

В первой строке ввода дано одно целое число  $n$  — количество ингредиентов у Зельды ( $1 \leq n \leq 2 \cdot 10^5$ ).

Во второй строке перечислены  $n$  целых чисел от 0 до  $10^9$  — номера этих ингредиентов.

### Формат выходных данных

Выведите единственное целое число — максимальный возможный номер секретного ингредиента.

### Примеры

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

### Замечание

В первом примере один из способов упорядочить ингредиенты выглядит как  $a = [5, 0, 1, 2, 1]$ . В этом случае мы получим  $b = [0, 1, 2, 3, 3]$ ,  $\text{mex}$  которого равен 4.

## Задача Н. Как провести время

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Когда Зельду бросили в тюрьму самозванцы, притворяющиеся королем и министром Хайрула, ей было долгое время нечем заняться. Поэтому она выписала последовательность из  $n$  натуральных чисел, после чего стала выполнять с ней следующие действия:

- Каждое нечетное действие Зельда находит минимальное число в последовательности  $m_1$ , а также второе по величине значение  $m_2 > m_1$ , после чего заменяет один из элементов последовательности, равный  $m_1$ , на  $m_2$ .
- Аналогично, каждое четное действие она находит максимальное число  $M_1$  и заменяет одно из его вхождений в последовательность на второй максимум  $M_2 < M_1$ .

Как только в последовательности останется не более двух различных чисел, Зельде надоест развлекаться с ней, и она наконец-то начнет предпринимать попытки сбежать из тюрьмы.

Чтобы ускорить этот процесс, определите, на каком ходу последовательность перестанет быть интересна Зельде, и какие значения в ней после этого останутся.

### Формат входных данных

В первой строке ввода дано целое число  $n$  — количество чисел в последовательности ( $1 \leq n \leq 10^5$ ).

Во второй строке перечислены  $n$  целых чисел  $a_i$  — сами элементы последовательности ( $1 \leq a_i \leq 10^5$ ).

### Формат выходных данных

В первой строке выведите номер хода, после которого Зельде перестанет быть интересна последовательность. Если последовательность с самого начала не является интересной, выведите 0.

Во второй строке выведите через пробел минимальное и максимальное числа в получившейся последовательности.

### Примеры

<code>stdin</code>	<code>stdout</code>
3 4 5 6	1 5 6
4 1 2 1 3	2 1 2
6 4 2 2 5 1 3	6 2 3

## Задача I. Фабрика эхо

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

Зельда нашла способ оптимизировать создание эхо объектов и создала конвейер по их производству.

Конвейер работает почти что по принципу обычной очереди, но также обладает и некоторыми дополнительными функциями: например, если где-то в очереди скопилось много эхо, которые Зельде уже не понадобятся, есть способ их быстро из очереди изъять.

Более подробно, конвейер в каждый момент времени может быть разделен на несколько сегментов, содержащих отрезки подряд идущих элементов очереди. Для работы с конвейером доступны следующие операции:

1. добавить элемент  $x$  в конец последнего сегмента очереди;
2. получить первый элемент первого сегмента очереди и удалить первый элемент **каждого** сегмента очереди; если какой-то сегмент после этого становится пустым, он удаляется, а его левый и правый соседи становятся соседними друг с другом;
3. разделить  $i$ -й по порядку сегмент очереди на первые  $j$  элементов и все оставшиеся; при такой операции  $i$ -й сегмент разбивается на два, а относительный порядок сегментов не меняется;
4. соединить все сегменты обратно в один сегмент, содержащий все элементы очереди.

Иными словами, порядок сохранившихся элементов в очереди не меняется, а при выполнении операции `pop` (2) помимо изъятия самого первого элемента очереди, также удаляются первые элементы каждого сегмента.

Для тестирования конвейера Зельда просит вас реализовать структуру данных, поддерживающую все эти операции.

### Формат входных данных

В первой строке ввода даны целые числа  $n$  и  $q$  — изначальное количество элементов в очереди и число запросов ( $1 \leq n, q \leq 10^5$ ).

Во второй строке ввода перечислены  $n$  целых чисел  $a_i$  — элементы очереди в ее начальном состоянии ( $1 \leq a_i \leq 10^9$ ).

Далее следует  $q$  запросов, по одному в каждой строке. Формат запросов в соответствии с их описанием в условии следующий: «`push x`», «`pop`», «`cut i j`» и «`restore`» ( $1 \leq x \leq 10^9$ ;  $1 \leq i \leq S$ , где  $S$  — количество непустых сегментов в данный момент;  $1 \leq j < C_S$ , где  $C_S$  — количество элементов в  $S$ -м сегменте).

Гарантируется, что все запросы корректны, в частности `pop` не вызывается при пустом первом сегменте, а `cut` не порождает пустые сегменты.

### Формат выходных данных

На каждый запрос `pop` выведите на отдельной строке вынимаемый первый элемент очереди.

Гарантируется, что хотя бы один такой запрос будет.

## Пример

стандартный ввод	стандартный вывод
7 8	1
1 9 239 144 25 16 777	9
pop	239
pop	144
cut 1 2	16
pop	777
restore	
pop	
pop	
pop	

## Замечание

В первом примере из условия очередь претерпевает следующие изменения:

1. [1, 9, 239, 144, 25, 16, 777];
2. [9, 239, 144, 25, 16, 777];
3. [239, 144, 25, 16, 777];
4. [239, 144], [25, 16, 777];
5. [144], [16, 777];
6. [144, 16, 777];
7. [16, 777];
8. [777];
9. [];

## Задача J. Великий комбинатор Зельда

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

But a serried ring of flashing steel Growltiger  
did surround

*Cats*

И вот Зельда зашла в новую локацию, и перед ней внезапно появилось  $n$  противников: засада Нулла, не иначе. Для удобства в бою Зельда дала каждому противнику свой номер, кроме этого она быстро выяснила вид каждого противника. Вид противника с номером  $i$  равен  $k_i$ , при этом у разных противников может быть один и тот же вид.

Поскольку на Зельду напали внезапно, она не смогла основательно подготовиться к сражению. Однако, благодаря наблюдениям за сражениями Линка, Зельда смогла вспомнить, что некоторые противники образуют между собой связи, делающие их уязвимыми к атакам. Значит, при правильной стратегии можно победить всех одним ударом!

К несчастью, Зельда не очень внимательно следила за сражениями Линка, поэтому смогла заметить лишь  $n - 1$  связь и тот факт, что любые два противника связаны напрямую или косвенно (по цепочке).

Из-за частых конфликтов с отцом принцесса не очень сильна в комбинированных атаках, которые поражают многих врагов за раз. А именно, она может выбрать два вида противников  $k_i$  и  $k_j$ , после чего выстрелить в какого-либо врага. Как только выстрел долетает до врага,

1. если этот враг не одного из двух выбранных видов, то с ним ничего не происходит;
2. иначе, этот враг оказывается повержен, а атака той же силы применяется ко всем связанным с ним врагам;
3. для каждого врага, который был атакован по цепочке связей, рекурсивно применяется тот же алгоритм.

Иными словами, за одну атаку Зельда может поразить множество связанных напрямую или косвенно врагов, каждый из которых принадлежит одному из выбранных ей до атаки видов.

Зельда хочет победить за наименьшее количество атак, так как она хочет поскорее прийти на помощь Линку. Она неплохо стреляет, но не уверена, какие виды противников выбирать для каждой атаки и кого ей нужно атаковать. Помогите Зельде с планом битвы, чтобы у неё точно получилось спасти Линка и весь Хайрул.

### Формат входных данных

В первой строке ввода дано одно число  $n$  — количество противников Зельды ( $1 \leq n \leq 2 \cdot 10^5$ ).

Во второй строке перечислены  $n$  целых чисел  $k_1, \dots, k_n$  — виды противников ( $1 \leq k_i \leq n$ ).

В  $i$ -й из следующих  $n - 1$  строках даны два целых числа  $v_i$  и  $u_i$ , означающие, что атака по противнику  $v_i$  перейдёт на противника  $u_i$  и наоборот ( $1 \leq v_i, u_i \leq n$ ;  $v_i \neq u_i$ ). Гарантируется, что такие связи косвенно связывают любую пару противников.

### Формат выходных данных

В первой строке необходимо вывести минимальное количество атак  $a$ .

В следующих  $a$  строках необходимо вывести список противников, пораженных каждой атакой. Строка, описывающая  $i$ -ю атаку, должна начинаться с числа  $n_i$  — количества поверженных противников, за которым через пробел должны быть перечислены  $n_i$  номеров поверженных противников от 1 до  $n$ .

Каждый враг должен быть выведен ровно один раз.

## Примеры

стандартный ввод	стандартный вывод
6 6 6 2 1 6 4 1 6 4 3 1 2 3 5 2 3	3 4 1 2 3 5 1 4 1 6
5 4 5 4 4 5 4 5 2 5 5 1 3 5	1 5 1 2 5 3 4

## Задача К. Священное бревно

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

Зельда хочет улучшить свойства своего Жезла Трифа и собирается использовать для этого дерево, растущее у Источника Силы. Для этого ей нужно отнести священное бревно мастеру-гному. Добравшись к дереву через весь Хайрул, она обнаружила, что это довольно большое дерево, и если она возьмет его с собой целиком, то оно превысит вместимость ее рюкзака на  $w$  единиц вместимости.

Где-то в своем арсенале Зельда нашла заклинание, которое на час превращает ее в бобра и теперь она собирается им наконец-то воспользоваться!

Дерево магическое (подвешенное), состоит из  $n$  вершин с корнем в вершине 1, а каждая вершина имеет свой вес  $a_i$ .

За одно действие Зельда-бобер может выбрать некоторую вершину дерева  $v$  и откусить ее со всеми ее *отростками* (вершина  $a$  является отростком вершины  $b$ , если кратчайший путь от вершины  $a$  до корня проходит через вершину  $b$ ).

Помогите Зельде-бобру использовать заклинание так, чтобы суммарный вес всех откушенных ею вершин был в точности равен  $w$ .

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $w$  — количество вершин в дереве и необходимый вес вершин для удаления ( $1 \leq n, w \leq 1000$ ).

Вторая строка входных данных содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  — веса вершин ( $1 \leq a_i \leq 100$ ).

В  $i$ -й из следующих  $n - 1$  строк даны два целых числа  $u_i$  и  $v_i$  — номера вершин, соединённых  $i$ -м ребром. Гарантируется, что данный граф является деревом ( $1 \leq u, v \leq n; u \neq v$ ).

### Формат выходных данных

Если с помощью описанных действий невозможно отгрызть вершины с суммарным весом  $w$ , выведите  $-1$ .

Иначе выведите сначала целое число  $k$  от 0 до  $n$  — количество превращений в бобра. Затем выведите  $k$  различных целых чисел от 1 до  $n$  — номера вершин, которые нужно отгрызть.

Если существует несколько подходящих последовательностей действий, выведите любую.

## Примеры

стандартный ввод	стандартный вывод
5 7 3 4 3 2 2 1 2 2 3 2 4 4 5	2 4 3
5 8 3 4 3 2 2 1 2 2 3 2 4 4 5	-1
5 14 3 4 3 2 2 1 2 2 3 2 4 4 5	1 1

## Замечание

В первом примере из условия Зельда-бобер отгрызает вершины 3 и 4, вместе с вершиной 4 она отгрызет и вершину 5, так как 5-я вершина является отростком вершины 4. Суммарный вес удаленных вершин 3, 4, 5 равен  $3 + 2 + 2 = 7$

## Задача L. Очередная карточная игра

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

У принцессы Зельды была колода из  $n$  карт. На каждой карте было написано целое число от 1 до  $n$ , причём изначально не было двух карт с одинаковыми числами на них. Затем она использовала силу волшебного Жезла Трифа и создала копию каждой карты.

К сожалению, карты упали на землю рубашками вверх, и теперь нужно снова собрать все карты по парам, иначе ими не получится пользоваться.

Просто перевернуть все карты было бы слишком скучно, поэтому Зельда решила выбирать по две карты и смотреть, совпали ли их значения. Если совпали, то она их откладывает в сторону, и выбирает следующую пару карт. Иначе она переворачивает эти карты обратно и выбирает пару карт заново.

Единственное, что Зельде известно изначально — если пронумеровать все лежащие на земле карточки от 1 до  $2n$ , то значение на карточке под номером  $a$  строго меньше, чем на карточке под номером  $b$ .

Несмотря на желание сделать процесс сортировки карт интересным, принцесса очень спешит, поэтому хочет закончить за не более чем  $2n - 1$  ход. Помогите ей в этом!

### Формат входных данных

В первой строке даны три целых числа  $n$ ,  $a$  и  $b$  — количество карточек, которые изначально были у Зельды, а также номера двух карточек, про значения которых известно, что первое меньше второго ( $2 \leq n \leq 50\,000$ ;  $1 \leq a, b \leq 2n$ ;  $a \neq b$ ).

После считывания ввода вашей программой запускается процесс взаимодействия с интерактором.

### Протокол взаимодействия

В процессе взаимодействия с интерактором ваша программа должна сообщать интерактору запросы в формате «?  $x$   $y$ », где  $x$  и  $y$  — два различных целых числа от 1 до  $2n$ , обозначающие номера карточек, которые Зельда должна перевернуть на очередном ходу.

В ответ интерактор в отдельной строке напечатает через пробел два целых числа от 1 до  $n$  — значения, написанные на этих карточках. Если значения совпали, эти карточки убираются и больше не должны упоминаться в запросах.

Если какой-то из сделанных вашей программой запросов будет некорректен, интерактор выведет «-1 -1» и завершится с вердиктом WA (Wrong Answer). Также если ваша программа превысит лимит в  $2n - 1$  запрос, интерактор завершится тем же образом.

Если же ваша программа сможет на  $2n - 1$  запрос или меньше перевернуть все карточки по парам с одинаковыми значениями, интерактор завершится с вердиктом OK. Во избежание получения вердиктов TL (Time Limit Exceeded) или IL (Idleness Limit Exceeded) ваша программа также должна завершаться с кодом возврата 0 после успешного ответа интерактора на последний запрос.

Также обратите внимание, что вывод каждого запроса должен завершаться переводом строки (символ '\n') и сбросом буфера вывода (`sys.stdout.flush()` в Python, `cout.flush()` в C++, `System.out.flush()` в Java и аналогичными методами в других языках).

## Примеры

стандартный ввод	стандартный вывод
2 1 4 1 1 2 2	1 3 2 4
3 5 3 3 2 3 3 1 2 2 2 1 1	3 5 3 4 1 5 2 5 1 6