

## Задача А. Сортировка листьев

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

Вам дано бинарное дерево с  $n$  вершинами. У него  $l$  листьев.

В этом дереве каждая вершина одного из двух типов:

- Лист. У этой вершины нет сыновей. В каждом листе записано одно целое число от 1 до  $l$ .
- Нелистовая вершина. У этой вершины два сына — левый и правый.

Гарантируется, что все числа, записанные в листьях, различные.

Вы можете делать следующую операцию сколько угодно раз: взять нелистовую вершину и поменять направление левого и правого сыновей (левый сын становится правым, а правый сын становится левым).

Найдите минимальное количество операций, которое нужно, чтобы значения в листьях были отсортированы по возрастанию в порядке обхода в глубину (при обходе мы идем сначала в левого сына, потом в правого). Если значения в листьях отсортировать невозможно, сообщите об этом.

### Формат входных данных

В первой строке находится единственное целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных. Их описание следует дальше, каждый набор начинается с новой строки.

В первой строке описания каждого набора входных данных находится единственное целое число  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — количество вершин бинарного дерева. Гарантируется, что сумма  $n$  по всем наборам не превосходит  $10^6$ .

Каждая из следующих  $n$  строк содержит два целых числа:  $l_i, r_i$  — информация об  $i$ -й вершине в  $i$ -й из этих строк. Если  $l_i = -1$ , то  $i$ -я вершина лист и  $r_i$  это число, записанное в ней. Иначе  $l_i$  и  $r_i$  это номера левого и правого сыновей, соответственно ( $1 \leq l_i, r_i \leq n, l_i \neq r_i$ ).

### Формат выходных данных

Для каждого набора входных данных:

Если отсортировать значения в листьях невозможно, выведите  $-1$ . Иначе выведите минимальное количество операций, которое для этого требуется.

### Пример

стандартный ввод	стандартный вывод
2	1
5	-1
3 5	
-1 2	
2 4	
-1 3	
-1 1	
7	
2 3	
4 5	
6 7	
-1 1	
-1 4	
-1 3	
-1 2	

## Задача В. Разбиение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	256 мегабайт

Ваня работает в поисковом отделе Очень Известной Компании. Каждый день его код обрабатывает миллионы запросов.

Программа, которую пишет Ваня, занимается обработкой запроса. А именно, он разбивает строку запроса на непересекающиеся подстроки-токены. Каждый из токенов должен быть подстрокой строки  $t$ . При этом, хотелось бы, чтобы токены в разбиении были большими (то есть токены отражали смысл поискового запроса). А именно, длина минимального токена должна быть максимальной.

Сегодня на сервер пришел поисковый запрос, состоящий из строки  $s$ , и Ваня код перестал работать, упав по Undefined behavior. К сожалению, ответить на запрос нужно прямо сейчас, поэтому Ваня просит вас помочь ему. Реализуйте его программу.

### Формат входных данных

На вход программе даются строки  $s$ ,  $t$  ( $1 \leq |s|$ ,  $|t| \leq 10^5$ ). Строки состоят из букв латинского алфавита, причем могут быть как заглавными, так и строчными.

### Формат выходных данных

Выведите максимально возможную длину минимальной по длине подстроки разбиения. Если искомого разбиения не существует, выведите «1».

### Пример

стандартный ввод	стандартный вывод
zachttop chtoproishoditzachto	1

## Задача С. Петербург?

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

– Это что за остановка –  
Бологое или Поповка? –  
А с платформы говорят:  
– Это город Ленинград.

---

«Вот какой рассеянный», Самуил Маршак

Пытаясь спастись от мира спортивного программирования, Алина сбежала на вокзал и уехала прочь на ночной электричке. Минуты медленно уплывали в даль, и уставшую девочку клонило в сон. Ей снился город-сказка, где не надо программировать, а можно гулять, мечтать и наслаждаться жизнью. Внезапно дождь из **интерактивных** задач разрушил эту идиллию.

Проснувшись и открыв окно, Алина задалась вопросом весьма философского свойства: «Где я?». С перрона потерявшейся девочке сообщили, что этот город, не похожий ни на что вокруг, представляет собой неориентированный граф на  $n$  вершинах и  $m$  ребрах. Сей невероятный факт, однако, нисколько не удивил Алину. Она давно мечтала побывать в одном таком городе — Петербурге. Его уникальной отличительной особенностью является то, что хотя бы **половина** его ребер — мосты (определение дано в конце условия). Так как никакие другие города Алине не интересны, она решила ограничиться расспросом находящихся на платформе эрудированных путешественников. Любой из них может по данной вершине  $v$  сообщить любое ещё не названное ребро, исходящее из нее, или же заявить об отсутствии таковых.

Алина неуверена в своих силах, поэтому попросила вас помочь ей определить, попала ли она в Петербург. Так как её поезд скоро продолжит свой путь, задать больше  $3n$  вопросов не получится.

Обратите внимание, что в графе **могут** присутствовать петли и кратные ребра.

### Протокол взаимодействия

В первой строке стандартного потока ввода даны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 100\,000$ ) — число вершин и ребер в графе соответственно.

Для того, чтобы узнать очередное ребро, исходящее из  $u$ -й вершины ( $1 \leq u \leq n$ ), нужно вывести «?  $u$ ». После этого ваша программа на вход получит целое число  $v$  ( $-2 \leq v \leq -1$  или  $1 \leq v \leq n$ ) —  $v = a + b - u$ , если существует ребро  $ab$ , которое инцидентно вершине  $u$  и **ещё не было названо**,  $-1$ , если такого ребра не существует и  $-2$ , если вы превысили допустимое число запросов. В последнем случае ваша программа должна немедленно завершиться, в ином случае жюри не гарантирует корректность полученного вами вердикта.

Вам разрешается задать не более  $3n$  вопросов.

Чтобы сообщить, что ответ найден, требуется вывести «! Yes» или «! No», в зависимости от того, является ли загаданный граф Петербургом. В случае положительного ответа выведите  $\lceil \frac{m}{2} \rceil$  строк, по два целых числа  $u_i$  и  $v_i$  в каждой ( $1 \leq u_i, v_i \leq n$ ), обозначающих, что ребро  $(u_i, v_i)$  является мостом. Любое ребро в приведенном списке должно встречаться не более одного раза (кратные ребра считаются различными).

Запрос на вывод ответа не входит в ограничение на  $3n$  запросов.

Не забывайте сбрасывать буфер после каждого запроса. Например, на языке C++ надо использовать функцию `fflush(stdout)` или вызов `cout.flush()`, на Java вызов `System.out.flush()`, на Pascal `flush(output)` и `stdout.flush()` для языка Python.

## Примеры

стандартный ввод	стандартный вывод
3 3	? 3
2	? 1
2	? 2
-1	? 1
3	? 1
-1	? 3
-1	! No
4 4	? 1
2	? 2
3	? 3
2	? 1
-1	? 3
4	? 3
-1	? 2
-1	? 4
-1	! Yes
	1 2
	3 4

## Замечание

В условии в примере взаимодействия вводимые и выводимые данные расположены для удобства восприятия в хронологическом порядке, при реальном взаимодействии никакие «лишние» переводы строк возникать не должны.

Ввод-вывод в примерах демонстрирует пример взаимодействия вашей программы с проверяющей системой.

В первом примере был загадан граф на трех вершинах с ребрами (1, 2), (2, 3) и (3, 1).

Во втором примере была загадан граф на четырех вершинах с ребрами (1, 2), (2, 3), (3, 4) и (2, 3).

Ребро, соединяющее вершины  $u$  и  $v$ , называется мостом, если после его удаления между вершинами  $u$  и  $v$  не существует пути.

## Задача D. Путь и циклы

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

Вам дан связный граф из  $n$  вершин без петель и кратных рёбер, такой что степень каждой вершины не менее 3, а также дано целое число  $1 \leq k \leq n$ .

У вас есть два варианта. Можно найти простой путь длины хотя бы  $\frac{n}{k}$ . Либо можно найти  $k$  простых по вершинам циклов с представителями, таких что:

- Длина каждого цикла не менее 3.
- Длина каждого цикла не кратна 3.
- В каждом цикле найдётся представитель — такая вершина, которая отсутствует во всех других **выведенных** циклах.

Если отсутствуют решения обоих вариантов — выведите  $-1$ .

### Формат входных данных

Первая строка содержит три числа  $n$ ,  $m$  и  $k$  ( $1 \leq k \leq n \leq 2.5 \cdot 10^5$ ,  $1 \leq m \leq 5 \cdot 10^5$ )

Последующие  $m$  строк содержат описание рёбер графа в виде пар вершин  $v$ ,  $u$  ( $1 \leq v, u \leq n$ ). Гарантируется, что  $v \neq u$  и что все  $m$  пар различны.

Гарантируется, что степень каждой вершины не менее 3.

### Формат выходных данных

Выведите PATH в первой строке, если выбрали первый вариант. Во второй строке выведите кол-во вершин в пути  $c$  ( $c \geq \frac{n}{k}$ ), а в третьей через пробел вершины в пути в порядке следования.

Выведите CYCLES в первой строке, если выбрали второй вариант. Далее нужно вывести **ровно**  $k$  циклов в следующем формате: в первой строке выведите кол-во вершин в цикле  $c$  ( $c \geq 3$ ), во второй строке выведите сам цикл в формате аналогичном пути, причем первая вершина во второй строке должна являться **представителем**

Выведите одно число  $-1$ , если отсутствуют оба решения.

Кол-во выведенных чисел не должно превосходить  $10^6$ . Гарантируется, что если существует хотя бы одно из решений (для одного из двух вариантов), то и существует вывод удовлетворяющий ограничению.

## Примеры

стандартный ввод	стандартный вывод
4 6 2 1 2 1 3 1 4 2 3 2 4 3 4	PATH 4 1 2 3 4
10 18 2 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 2 3 3 4 2 4 5 6 6 7 5 7 8 9 9 10 8 10	CYCLES 4 4 1 2 3 4 7 1 5 6

## Задача E. Евклид

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

В этой задаче от вас требуется считать gcd на отрезке (запросы «0 l r») и прибавлять арифметическую прогрессию  $k, 2k, \dots$  на отрезке (запросы «1 l r k»).

### Формат входных данных

В первой строке вводятся числа  $n, q$  — размер массива и число запросов. Оба числа не превышают  $10^5$ .

В следующей строке вводится массив из  $n$  натуральных чисел. В следующих  $q$  строках даются запросы. Гарантируется, что параметр  $k$  запроса больше нуля, а также и  $k$ , и элементы массива не превосходят  $2 \cdot 10^8$ .

### Формат выходных данных

Выведите на каждый get-запрос ответ в отдельной строке.

### Пример

стандартный ввод	стандартный вывод
8 3	4
2 8 12 24 66 33 21 7	2
0 2 4	
1 1 4 2	
0 2 4	

## Задача F. Жемчуг

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

Со стародавних времён в поморских деревнях рукодельницы вышивали жемчугом на прямоугольных полотенцах, состоящих из одинаковых клеток. Вышивка начиналась с пришивания жемчужины к полотенцу в центре одной из клеток. Чтобы пришить новую жемчужину, рукодельница делала стежок из клетки, уже содержащей жемчужину, в соседнюю с ней по горизонтали или вертикали свободную клетку. Новая жемчужина пришивалась в центре клетки на конце стежка. Этот процесс повторялся, пока не заканчивались жемчужины.

Одно из таких праздничных полотенеч находится в музее. К сожалению, некоторые части узора были утеряны, но описание полотенца сохранилось. Дирекция музея планирует восстановить один из прямоугольных фрагментов полотенца, но не ещё не решила какой именно. Затраты на восстановление фрагмента зависят от количества связанных частей узора, попавших на этот фрагмент. Часть узора считается связанной, если от любой её жемчужины можно по стежкам перейти к любой другой её жемчужине, не выходя за границы фрагмента. Дирекция всегда относит любые две жемчужины, между которыми можно перейти по стежкам, к одной и той же связанной части узора.

Требуется написать программу, вычисляющую количество связанных частей узора для каждого из заданных фрагментов.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq 150000$ ) — размеры полотенца в клетках по горизонтали и вертикали.

Вторая строка содержит два числа  $n$  и  $q$  ( $1 \leq n, q \leq 100000$ ) — количество жемчужин в узоре и количество фрагментов соответственно.

Следующие  $(n-1)$  строк содержат описания стежков. Каждый стежок имеет один из следующих видов:

- «h x y» означает, что клетки с координатами  $(x; y)$  и  $(x + 1; y)$  содержат жемчужины, соединённые горизонтальным стежком ( $1 \leq x \leq a - 1; 1 \leq y \leq b$ );
- «v x y» означает, что клетки с координатами  $(x; y)$  и  $(x; y + 1)$  содержат жемчужины, соединённые вертикальным стежком ( $1 \leq x \leq a; 1 \leq y \leq b - 1$ ).

Следующие  $q$  строк описывают фрагменты. Каждое описание содержит четыре целых числа  $x_1, y_1, x_2$  и  $y_2$  — координаты левой нижней и правой верхней клетки фрагмента ( $1 \leq x_1 \leq x_2 \leq a; 1 \leq y_1 \leq y_2 \leq b$ ).

### Формат выходных данных

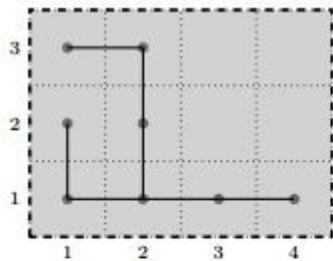
Выходные данные должны содержать  $q$  строк, где  $i$ -я строка содержит количество связанных частей узора в  $i$ -м фрагменте.



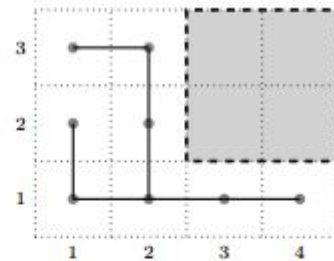
## Пример

стандартный ввод	стандартный вывод
4 3	1
8 4	0
v 1 1	1
h 1 1	2
h 2 1	
v 2 1	
v 2 2	
h 1 3	
h 3 1	
1 1 4 3	
3 2 4 3	
3 1 3 1	
1 2 3 3	

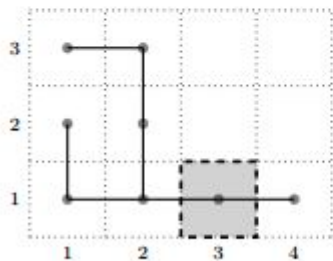
## Замечание



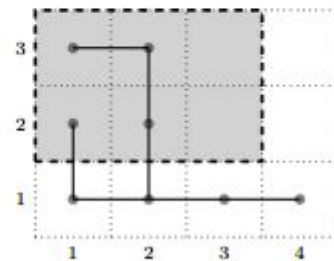
$$x_1 = 1, y_1 = 1, x_2 = 4, y_2 = 3$$



$$x_1 = 3, y_1 = 2, x_2 = 4, y_2 = 3$$



$$x_1 = 3, y_1 = 1, x_2 = 3, y_2 = 1$$



$$x_1 = 1, y_1 = 2, x_2 = 3, y_2 = 3$$