

Задача А. Кто убил Марка?

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

Зачёркивал, стирал, неустанно толстел
блокнот

Марк

У Марка есть изначально пустой граф на n вершинах.

У каждой вершины есть вес — неотрицательное целое число.

Также у Марка есть m троек целых чисел (a_i, b_i, s_i) , где $1 \leq a_i \neq b_i \leq n$, и $s_i \geq 0$.

После чего Марк запускает следующий процесс:

- Если не существует такого i , что вершины a_i и b_i лежат в разных компонентах связности графа и суммарный вес вершин в компонентах связности a_i и b_i не меньше s_i , то процесс завершается.
- Если же такое i есть, то выбирается наименьшее такое i , и в граф добавляется ребро между вершинами a_i и b_i , и в блокнот записывается число i , после чего процесс повторяется уже на большем графе.

Однако в результате загадочных обстоятельств десятилетней давности Марк пропал вместе со своим блокнотом. Вам необходимо восстановить числа, записанные в блокноте.

Формат входных данных

В первой строке входных данных дано два целых положительных числа n и m ($1 \leq n, m \leq 3 \cdot 10^5$) — количество вершин в графе и количество троек.

Вторая строка содержит n целых неотрицательных чисел w_i ($0 \leq w_i \leq 10^6$) — веса вершин графа.

Последующие m строк описывают тройки. Каждая строка содержит по три целых неотрицательных числа a_i, b_i и s_i ($1 \leq a_i \neq b_i \leq n$, $0 \leq s_i \leq 10^6$) — описание очередной тройки.

Формат выходных данных

В первой строке выведите одно целое число k — количество чисел, записанных в блокноте после окончания процесса.

Во второй строке выведите k целых чисел, которые записаны в блокноте (в правильном порядке).

Примеры

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

Задача В. Ехаб и странная формула веса

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Дано дерево, состоящее из n вершин. У каждой вершины u есть вес a_u . Гарантируется, что в дереве есть только одна вершина с минимальным весом. У каждой вершины u (кроме вершины с минимальным a_u), есть сосед v , такой, что $a_v < a_u$.

Вам необходимо составить дерево с минимальным весом w , который считается следующим образом:

- Для каждой вершины u , $deg_u \cdot a_u$ прибавляется к w (deg_u обозначает степень вершины u).
- Для каждого ребра $\{u, v\}$, $\lceil \log_2(dist(u, v)) \rceil \cdot \min(a_u, a_v)$ прибавляется к w , где $dist(u, v)$ обозначает количество ребер на пути из u в v в данном дереве.

Формат входных данных

В первой строке записано одно целое число n ($2 \leq n \leq 5 \cdot 10^5$), количество вершин в дереве.

Во второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), веса вершин дерева.

Далее следует $n-1$ строка, в каждой из которых записаны по два целых числа u и v ($1 \leq u, v \leq n$).

Это означает, что между u и v есть ребро в данном дереве.

Формат выходных данных

Выведите одно целое число: минимальное возможное значение w .

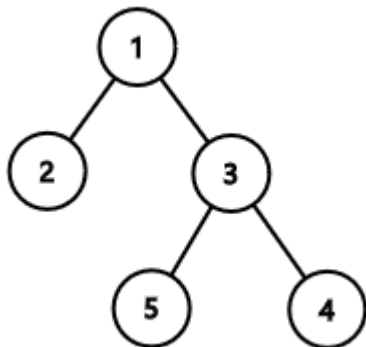
Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 2 1 3	7
5 4 5 3 7 8 1 2 1 3 3 4 4 5	40

Замечание

В первом примере данное дерево исходно имеет минимальное значение w .

Во втором примере оптимальное дерево имеет следующий вид:



Задача С. Странный прибор

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

Это интерактивная задача.

Вася обожает решать загадки и разгадывать головоломки. На этот раз он нашел странный прибор и хочет выяснить принцип его работы.

Этот прибор зашифрован с помощью дерева (связного неориентированного графа без циклов), состоящего из n вершин, пронумерованных целыми числами от 1 до n . Чтобы решить головоломку надо угадать это дерево.

К счастью прибор умеет выполнять одну операцию, исходя из которой надо разгадать его шифр. Можно ввести в прибор последовательность d_1, d_2, \dots, d_n целых неотрицательных чисел. На приборе есть n лампочек, i -я из которых отвечает за i -ю вершину дерева-шифра. Для всех i из этих лампочек i -я загорится, если существует такая вершина дерева-шифра с номером $j \neq i$, что $dist(i, j) \leq d_j$. Здесь $dist(i, j)$ обозначает расстояние между вершинами i и j в дереве-шифре, то есть количество ребер в простом пути между вершинами i и j .

Вася хочет за ≤ 80 операций с прибором решить головоломку и угадать дерево-шифр. Помогите ему!

Протокол взаимодействия

В начале вашей программе вводится единственное целое число n — количество вершин в дереве-шифре прибора ($2 \leq n \leq 1000$).

Далее вы можете выполнять операции в следующем формате. Сначала выведите символ “?” (без кавычек) и за ним n целых чисел d_1, d_2, \dots, d_n , разделенных пробелами. Заметьте, что в операциях для всех i должно быть выполнено неравенство $0 \leq d_i < n$. В ответ будет выведена строка s длины n , состоящая из символов “0” и “1” (без кавычек). Для всех i символ s_i будет равен “0”, если у прибора не включилась лампочка, соответствующая i -й вершине дерева-шифра и “1”, иначе.

После нескольких запросов вы должны вывести угаданное дерево. Для этого в первой строке выведите единственный символ “!” (без кавычек). В следующих $n - 1$ строках выведите по 2 целых числа a_i, b_i — номера вершин, соединяющих i -е ребро дерева. Выведенные числа должны удовлетворять условиям $1 \leq a_i, b_i \leq n$ и $a_i \neq b_i$. Эти ребра должны образовывать дерево, совпадающее с загаданным. Выводить ребра можно в любом порядке. После этого ваша программа должна завершиться.

Гарантируется, что в каждом тесте дерево-шифр будет зафиксировано заранее и не будет меняться в зависимости от выполняемых операций.

Ваша программа может выполнить от 0 до 80 операций с прибором и после этого сообщить дерево, совпадающее с загаданным.

Если ваша программа сделает больше 80 операций, то она может получить любой вердикт, потому что будет считывать данные из закрытого потока ввода. Также, если ваша программа сделает операцию или выведет ответ в неверном формате, она может получить любой вердикт. **Будьте внимательны.**

Не забудьте сбрасывать буфер вывода после того, как выведете данные для операции или ответ.

Чтобы сбросить буфер вывода вы можете использовать:

- `fflush(stdout)` в C++.
- `System.out.flush()` в Java.
- `stdout.flush()` в Python.
- `flush(output)` в Pascal.
- Прибегните к документации других языков.

Пример

стандартный ввод	стандартный вывод
5	? 0 0 0 0 0
00000	? 1 1 2 0 2
11011	? 0 0 0 1 0
11100	? 0 1 0 0 1
10010	!
	4 2
	1 5
	3 4
	4 1

Замечание

Таблица попарных расстояний между вершинами выглядит так:

	1	2	3	4	5
1	0	2	2	1	1
2	2	0	2	1	3
3	2	2	0	1	3
4	1	1	1	0	2
5	1	3	3	2	0

- Если сделать операцию, в которой $d = [0, 0, 0, 0, 0]$, то ни одна лампочка не загорится, потому что $dist(i, j) > 0$ при всех $i \neq j$.
- Если сделать операцию, в которой $d = [1, 1, 2, 0, 2]$, то загорятся все лампочки, кроме лампочки, соответствующей вершине дерева-шифра с номером 3. Например, лампочка, соответствующая вершине с номером 1 загорится, потому что $dist(1, 5) = 1 \leq 2 = d_5$.
- Если сделать операцию, в которой $d = [0, 0, 0, 1, 0]$, то загорятся все лампочки, кроме лампочек, соответствующих вершинам дерева-шифра с номерами 4 и 5.
- Если сделать операцию, в которой $d = [0, 1, 0, 0, 1]$, то загорятся только лампочки, соответствующие вершинам дерева-шифра с номерами 1 и 4.

Задача D. Погоня в метро

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Это интерактивная задача. Параллельно с выполнением вашего решения жюри запускает проверяющую программу с которой вы обмениваетесь сообщениями через стандартный ввод и вывод. Подробнее о протоколе взаимодействия написано ниже. Также в конце условия вы можете посмотреть корректные примеры взаимодействия с проверяющей программой на разных языках программирования.

В прекрасной Метрополии будущего необходимость в машинистах, управляющих поездами метро, отпала. Благодаря развитию технологий, их заменил искусственный интеллект (ИИ). К сожалению, в один прекрасный день опасения писателей-фантастов сбылись — ИИ взбунтовался, и теперь где-то в метро ездит неуправляемый поезд. Страшно представить, чем это грозит городской транспортной системе! Ваша задача состоит в том, чтобы найти поезд в сложной системе метро и остановить неуправляемый ИИ.

В целях безопасности все остальные поезда были отправлены в депо, а все ветки, кроме той, на которой находится неконтролируемый поезд, были перекрыты, поэтому на данный момент метро Метрополии представляет из себя одну ветку (обычную прямую ветку без самопересечений) из n станций, последовательно пронумерованных от 1 до n , ровно на одной из которых находится поезд. Для поимки неуправляемого поезда вам требуется определить номер этой станции, после чего на путях будут установлены искусственные заграждения и поезд будет пойман.

Для определения нужной станции диспетчер Сара одолжила вам устройство, позволяющее вам выбрать произвольные числа l и r ($l \leq r$), после чего оно проверит, верно ли, что поезд находится на станции с номером между l и r . К сожалению, для перезарядки устройства требуется k минут (и вы используете его как только перезарядка завершается), поэтому между двумя применениями поезд может перебраться из той станции, где он сейчас находится, в любую станцию с номером отличающимся не более чем на k . Формально, если при некотором применении устройства поезд находился на станции x , то при следующем применении он может находиться на любой станции y , такой что $\max(1, x - k) \leq y \leq \min(n, x + k)$. При этом поезд не знает, что вы пытаетесь его поймать и совершает все перемещения согласно некоторому заранее составленному им плану.

В процессе изучения устройства вы выяснили, что оно было сделано очень давно, и сможет выдержать не более чем q использований, после чего оно сломается, а ваша задача будет считаться проваленной.

Сможете ли вы найти станцию, на которой находится поезд, за не более чем q использований устройства?

Протокол взаимодействия

При запуске решения на вход подаются три целых числа n ($1 \leq n \leq 10^{18}$), k ($0 \leq k \leq 10$) и q ($4500 \leq q \leq 15\,000$).

Чтобы использовать устройство, вы должны вывести через пробел два числа l и r ($1 \leq l \leq r \leq n$). В ответ на это вы получите либо строку «Yes», если между станциями с номерами l и r находится поезд, либо строку «No» иначе. Если $l = r$ и вы получили ответ «Yes», это значит, что вы точно определили станцию, на которой находится поезд, и ваша программа после этого должна немедленно завершиться.

Если ваша программа за q запросов не смогла точно определить станцию, на которой находится поезд, программа должна также немедленно завершиться, в противном случае вердикт тестирующей системы может быть любым.

После каждого запроса необходимо вывести перевод строки и сбросить буфер вывода — для этого используйте команды `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java. В точности соблюдайте формат взаимодействия с системой.

Пример

стандартный ввод	стандартный вывод
10 2 15000	
Yes	3 4
No	3 3
Yes	2 2

Замечание

В первом тесте поезд изначально находился на станции 3, после первого использования устройства переместился на станцию 2, а после второго — остался на месте.

Задача Е. 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, несколько 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 F. Количество прямоугольных треугольников

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

На плоскости отмечены n различных точек. Необходимо отметить еще одну точку (не совпадающую ни с одной из уже имеющихся) так, чтобы максимизировать количество прямоугольных треугольников, вершины которых — отмеченные точки, а катеты параллельны осям координат.

Формат входных данных

Первая строка входных данных содержит одно целое положительное число n ($1 \leq n \leq 2 \cdot 10^5$) — количество отмеченных точек.

Каждая из последующих n строк содержит по два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты отмеченных точек.

Формат выходных данных

Выведите одно число — максимальное количество прямоугольных треугольников, которые можно получить, если отметить еще ровно одну точку.

Примеры

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

Задача G. Игра пингвина

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Обратите внимание: эта задача интерактивная.

Пингвин Ксорий недавно придумал новую игру. У него есть n сосулек, занумерованных целыми числами от 1 до n . Каждая сосулька имеет температуру — целое число от 1 до 10^9 . **Ровно две** из этих сосулек особые: их температура равна y , в то время как температура всех остальных сосулек равна $x \neq y$. Вам нужно найти две особые сосульки. Вы можете не более **19** раз выбрать некоторое *непустое* подмножество сосулек и спросить у пингвина, чему равно побитовое исключающее ИЛИ (XOR) температур сосулек из этого подмножества.

Вам необходимо найти особые сосульки.

Формат входных данных

Первая строка содержит три целых числа n, x, y ($2 \leq n \leq 1000, 1 \leq x, y \leq 10^9, x \neq y$) — количество сосулек, температура обычных сосулек и температура особых сосулек.

Формат выходных данных

Чтобы сообщить пингвину свой ответ, выведите символ «!» (без кавычек), затем два целых числа p_1, p_2 ($p_1 < p_2$) — номера особых сосулек **в порядке возрастания**. Обратите внимание, что «!» и p_1 должны быть разделены пробелом; номера также должны быть разделены пробелом. После того, как вы сообщили ответ, ваша программа должна немедленно завершиться.

Протокол взаимодействия

Чтобы задать вопрос, выведите символ «?» (без кавычек), число c ($1 \leq c \leq n$), а затем c различных чисел p_1, p_2, \dots, p_c ($1 \leq p_i \leq n$) — номера сосулек, про которые вы хотите узнать информацию. Обратите внимание, что «?» и c должны быть разделены пробелом; номера также должны быть разделены пробелами.

После того, как вы задали вопрос, необходимо считать одно число — ответ на вопрос.

Обратите внимание, что вы можете задать не более **19** вопросов. В случае, если вы зададите больше 19 вопросов или зададите хотя бы один некорректный вопрос, ваше решение получит вердикт «Неправильный ответ».

Если в какой-то момент ваша программа считывает -1 как ответ, она должна немедленно завершиться (например, вызовом `exit(0)`). Вы получите вердикт «Неправильный ответ», и это будет означать, что вы задали больше 19 вопросов или задали некорректный вопрос. Если вы проигнорируете это, то можете получить любой вердикт, так как ваша программа продолжит читать из закрытого потока ввода.

Выше решение получит вердикт «Решение зависло», если вы не будете ничего выводить или забудете сделать операцию `flush` после вывода вопроса или ответа.

Чтобы выполнить операцию `flush`, можете использовать (сразу после вывода чисел и перевода строки):

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

Пример

стандартный ввод	стандартный вывод
4 2 1	? 3 1 2 3
2	? 1 1
1	? 1 3
1	! 1 3

Замечание

Для первого вопроса ответ равен $1 \oplus 2 \oplus 1 = 2$.

Для второго и третьего вопросов ответ равен 1, следовательно, особые сосульки имеют номера 1 и 3.

Задача Н. Связность плоского графа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 512 мегабайт

Дан плоский граф. Поступают запросы двух видов:

- $- a_i b_i$ — удалить ребро между вершинами a_i и b_i и вывести количество компонент связности в графе после этой операции.
- $? a_i b_i$ — вывести, правда ли, что вершины a_i и b_i в данный момент лежат в одной компоненте связности.

Формат входных данных

В первой строке входных данных даны два целых положительных числа n и q ($1 \leq n \leq 10^5$, $1 \leq q \leq 2 \cdot 10^5$) — количество вершин в графе и количество запросов.

Каждая из последующих n строк начинается с числа k_i ($1 \leq k_i \leq n-1$), равного степени вершины i в изначальном графе. Далее идут k_i чисел v_{ij} ($1 \leq v_{ij} \leq n$), обозначающие соседей вершины i в порядке обхода по часовой стрелке.

Последующие q строк задают запросы. Обратите внимание, что числа в запросах закодированы. Если ответ на предыдущий запрос равен $last$, то запрос необходимо произвести на числах $a_i \oplus last$ и $b_i \oplus last$ (считайте, что изначально $last = 0$, \oplus — операция побитового исключающего или).

Формат выходных данных

На каждый запрос первого типа выведите одно число — количество компонент связности.

На каждый запрос второго типа выведите одно число: «1», если вершины a_i и b_i лежат в одной компоненте связности, и «0» в противном случае.

Пример

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

Задача I. Alien Invasion Online

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

This is an interactive problem.

Recently, aliens from planet *Nibiru*, also known as *Not-Taking-It*, have decided to invade the Earth. As the chief of the Nibiru Special Forces headquarters, you are planning to land the first group of spaceships in Byteland, the most peaceful country of the Earth.

You know that the territory of Byteland has a shape of a polygon with n vertices on a plane (yes, Byteland is not very large, so we can view the surface around Byteland as part of the plane). Each vertex of this polygon represents a town, and each side represents a part of the border of Byteland. Of course, the border cannot intersect itself, and two sides of the polygon can have a common point only if they are neighboring (in this case, there is exactly one common point).

Your task is to find the exact area of Byteland in order to calculate the number of spaceships to be located there. However, you do not know the exact shape of Byteland, since it would take too much time to discover. Fortunately, the Nibiru Intelligence Agency has sent spies in each of the n towns of Byteland. Each spy has a huge signal lamp. You also know that the spies are numbered 1 through n such that the spies with neighboring numbers (in other words, numbers that differ either by 1 or by $n - 1$) are located in towns that are directly connected by a part of the border.

You can perform the following *special operation*. At first, you tell each spy to switch his lamp either on or off. Then, you send an astronaut to the Earth, she sees all lighted lamps and tells you the least area of a convex set containing all these lamps (in other words, the area of the convex hull of these lamps). As it is difficult to keep orientation in space, the astronaut can tell nothing else about positions of the towns.

Note that if you perform a *special operation* more than $\frac{n(n-1)}{2}$ times, the Bytelandian police will notice something strange and the invasion will fail. Find the exact area of Byteland without making too much noise!

Протокол взаимодействия

At first, you get one integer n ($3 \leq n \leq 200$), the number of vertices of the polygon representing Byteland.

Then you make at most $\frac{n(n-1)}{2}$ *special operations*. A query describing the special operation should have the form of two lines:

? k

$a_1 a_2 \dots a_k$

where k ($3 \leq k \leq n$) is the number of spies that have to switch on their lamps and a_1, \dots, a_k ($1 \leq a_i \leq n$, $a_i \neq a_j$ for $i \neq j$) are the indices of towns where these spies are located. All spies in all other towns will switch off their lamps.

After making a query, you should read one line with a single non-negative number from the input: the area of the convex hull built on lamps that are lit up (in square meters).

When you think you know the area x of Byteland, print

! x

on a single line. Printing the answer does not count as a special operation. The answer should be accurate, neither any errors nor leading zeros are allowed. After that, your program should terminate gracefully.

Remember to put a newline character and flush the output buffer after making a query or printing an answer.

It is guaranteed that the polygon does not change during the program execution, and the area of the polygon is strictly greater than zero. Also, it is guaranteed that there exists a Cartesian coordinate system on a plane containing Byteland such that both coordinates of all vertices of the polygon (in meters) are integers in the range $[-10^9, 10^9]$.

Пример

standard input	standard output
3	? 3
	1 2 3
0.5	! 0.5

Замечание

For the sample test, the Bytelandian towns are located in points $(0,0)$, $(0,1)$ and $(1,0)$ for some Cartesian coordinate system.

Note that blank lines in sample input and output are printed only for clarity and do not exist in reality.

Задача J. Milliarium Aureum Online

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 0.5 seconds
Ограничение по памяти: 256 megabytes

Roman Empire is vast and, as everybody knows, all roads lead to Rome!

Following the trail of the Last Centurion, professor Melody Song found an ancient map of Roman roads. The exact position of Rome was forgotten long ago, and Melody wants to recover it from the map. There are n cities on the map, numbered from 1 to n , and m roads. Each road is marked as either *minor* or *major*.

Major roads were used to travel to Rome and formed a spanning tree, and minor roads were used as alternatives or to travel between other cities. Each road had some fixed width. It is known that major roads were very wide: if we consider two paths from Rome to any other city, an arbitrary path P and a simple path Q using the major roads, the thinnest road on path P could not be wider than the thinnest road on path Q .

The map found by Melody contains information on every road in Roman Empire, namely its type t and width w . Your task is to help her determine which cities may correspond to Rome according to the map.

Формат входных данных

The first line of input contains two integers n and m ($1 \leq n, m \leq 10^5$).

Each of the next m lines contains four integers, t_i, u_i, v_i , and w_i , which describe a bidirectional road from city u_i to city v_i with type t_i and width w_i ($1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^9$). The type is $t_i = 0$ for minor roads and $t_i = 1$ for major roads.

There may be multiple roads between the same pair cities, and there may be roads connecting a city to itself. It is guaranteed though that there is exactly one simple path via major roads between each pair of cities.

Формат выходных данных

On the first line, output an integer k which is number of cities which may be Rome.

On the second line, output k integers **in ascending order** which are the numbers of those cities.

It is guaranteed that there is at least one such city.

Примеры

standard input	standard output
4 6 1 1 2 2 1 1 3 2 1 1 4 2 0 2 3 3 0 3 4 3 0 4 2 3	1 1
3 3 0 2 3 1 1 1 2 2 1 1 3 2	3 1 2 3

Задача К. Нелёгкие подпоследовательности [на 10]

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

Дано n строк из маленьких английских букв. Рассмотрим все перестановки этих строк. Определите, для скольких из них строка, полученная конкатенацией строк, переставленных в соответствии с перестановкой, имеет чётное число различных подпоследовательностей.

Формат входных данных

Первая строка содержит целое число n ($2 \leq n \leq 20$).

Следующие n строк содержат строки из маленьких английских букв. Длины каждой строки не превосходят 10^5 .

Формат выходных данных

Выведите единственное число — ответ.

Примеры

стандартный ввод	стандартный вывод
2 a ba	1
3 a a baa	4

Замечание

Рассмотрим первый пример.

У строки "baa" = "ba" + "a" различных подпоследовательностей 6: пустая, "a", "b", "aa", "ba", "baa".

У строки "aba" = "a" + "ba" различных подпоследовательностей 7: пустая, "a", "b", "aa", "ab", "ba", "aba".

Поэтому ответ 1.

Задача L. Toll

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Простите, что условие на английском. Но оно не очень сложное. ;)

Happyland can be described by a set of N towns (numbered 1 to N) initially connected by M bidirectional roads (numbered 1 to M). Town 1 is the central town. It is guaranteed that one can travel from town 1 to any other town through these roads. The roads are toll roads. A user of the road i has to pay a toll fee of c_i cents to the owner of the road. It is known that all of these c_i 's are distinct. Recently, K additional new roads are completed and they are owned by a billionaire Mr. Greedy. Mr. Greedy can decide the toll fees (not necessarily distinct) of the new roads, and he has to announce the toll fees tomorrow.

Two weeks later, there will be a massive carnival in Happyland! Large number of participants will travel to the central town and parade along the roads. A total of p_j participants will leave from town j and travel toward the central town. They will only travel on a set of selected roads, and the selected roads will be announced a day before the event. By an old tradition, the roads are to be selected by the richest person in Happyland, who is Mr. Greedy. Constrained by the same tradition, Mr Greedy must select a set of roads that minimizes the sum of toll fees in the selected set and yet at the same time allow anyone to travel from town j to town 1 (hence, the selected roads form a "minimum spanning tree" where the toll fees are the weights of the corresponding edges). If there are multiple such sets of roads, Mr. Greedy can select any set as long as the sum is minimum.

Mr. Greedy is well-aware that the revenue he received from the K new roads does not solely depend on the toll fees. The revenue from a road is actually the total fee collected from people who travel along the road. More precisely, if p people travel along road i , the revenue from the road i is the product $c_i \cdot p$. Note that Mr. Greedy can only collect fees from the new roads since he does not own any of the old roads.

Mr. Greedy has a sneaky plan. He plans to maximize his revenue during the carnival by manipulating the toll fees and the roads selection. He wants to assign the toll fees to the new roads (which are to be announced tomorrow), and select the roads for the carnival (which are to be announced a day before the carnival), in such a way that maximizes his revenue from the K new roads. Note that Mr. Greedy still has to follow the tradition of selecting a set of roads that minimizes the sum of toll fees.

You are a reporter and want to expose his plan. To do so, you have to first write a program to determine how much revenue Mr. Greedy can make with his sneaky plan.

Формат входных данных

The first line contains three space-separated integers N , M and K ($1 \leq N \leq 10^5$, $1 \leq M \leq 3 \cdot 10^5$, $1 \leq K \leq 20$).

The next M lines describe the initial M roads. The i -th of these lines contains space-separated integers a_i , b_i and c_i ($1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10^6$, all c_i are distinct), indicating that there is a bidirectional road between towns a_i and b_i with toll fee c_i .

The next K lines describe the newly built K additional roads. The i -th of these lines contains space-separated integers x_i and y_i ($1 \leq x_i \leq y_i$), indicating that there is a new road connecting towns x_i and y_i .

The last line contains N space-separated integers, the j -th of which is p_j ($1 \leq p_j \leq 10^6$), the number of people from town j traveling to town 1.

It's guaranteed that between any two towns, there is at most one road (including newly built ones).

Формат выходных данных

Your program must write to the standard output a single integer, which is the maximum total revenue obtainable

Пример

стандартный ввод	стандартный вывод
5 5 1 3 5 2 1 2 3 2 3 5 2 4 4 4 3 6 1 3 10 20 30 40 50	400

Замечание

In this sample, Mr. Greedy should set the toll fee of the new road (1,3) to be 5 cents. With this toll fee, he can select the roads (3,5), (1,2), (2,4) and (1,3) to minimize sum of toll fees, which is 14 cents. 30 people from town 3 and 50 people from town 5 will pass through the new road to town 1 and hence he can collect an optimal revenue of $(30 + 50) \times 5 = 400$ cents. If, on the other hand, the toll fee of the new road (1,3) is set to be 10 cents. Now, constrained by the tradition, Mr. Greedy must select (3,5), (1,2), (2,4) and (2,3) as this is the only set that minimizes the sum of toll fees. Hence, no revenue will be collected from the new road (1,3) during the carnival.