

Задача А. Встреча у фонтана

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Кира договорилась с подругой Аней встретиться в центре парка у фонтана через T минут. Расстояние до места встречи с Аней n метров. Путь до места встречи сначала проходит по улицам, а затем по парку. Кира поняла, что может не успеть прийти вовремя, если пойдёт пешком, поэтому взяла напрокат электросамокат.

Кира проехала по улицам на самокате m метров со скоростью x м/с до парка. Парк — пешеходная зона, в которой ограничена скорость электросамокатов, поэтому сразу после въезда в парк Кира обнаружила, что скорость самоката снизилась до y м/с.

Определите, успеет ли Кира на встречу с Аней вовремя, и если не успеет, то на сколько минут она опоздает.

Формат входных данных

В первой строке ввода даны два целых числа n ($500 \leq n \leq 10000$) и T ($1 \leq T \leq 60$) — расстояние в метрах от места старта Киры на электросамокате до места встречи и через сколько минут назначена встреча.

Во второй строке дано единственное целое число m ($100 \leq m \leq 9000$, $m \leq n$) — расстояние в метрах от места старта Киры до входа в парк.

В третьей строке даны два целых числа x и y — ($3 \leq y < x \leq 9$) — скорость в м/с электросамоката на улице и после въезда парк, соответственно.

Формат выходных данных

Если Кира успеет вовремя, выведите единственное число — 0.

Если Кира опоздает, то выведите целое число L — сколько минут Аня будет ждать Киру. Это число необходимо округлить вверх.

Пример

стандартный ввод	стандартный вывод
5000 10 2500 8 4	6

Задача В. Взлом навигатора

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

У студента Дениса сегодня очередная пересдача экзамена по информационной безопасности. Он поздно проснулся и опаздывал, поэтому заказал такси.

Город, в котором живет Денис, можно представить в виде n перекрёстков, которые соединены m двусторонними дорогами. Про каждую дорогу известно, за какое время можно по ней проехать. Дом Дениса расположен на перекрёстке с номером s , а университет — возле перекрёстка с номером t . Как только такси доедет до перекрёстка с университетом, Денис немедленно покинет машину и побежит на пересдачу.

Однако не все так просто — как раз сегодня хакеры взломали навигатор в такси. Обычно навигатор помогает следовать по кратчайшему пути до пункта назначения: на каждом перекрестке навигатор показывает, по какой дороге нужно проехать, чтобы добраться до пункта назначения как можно быстрее. Если таких дорог несколько, навигатор может указать любую. Благодаря взлому хакеры могут заставить навигатор не более одного раза во время движения показать в качестве маршрута любую дорогу, на усмотрение хакеров.

Денис не запомнил маршрут, но ему известно время L , которое заняла поездка до университета. Оказалось, что преподаватель знает о хакерской атаке, и в качестве задания для пересдачи он просит Дениса посчитать количество различных способов, которыми он мог добраться до университета за время L .

Помогите Денису посчитать количество путей от его дома до университета, перемещение по которым занимает время L , таких, что такси может поехать по этому пути с учетом взлома навигатора. Поскольку путей может быть очень много, выведите остаток от деления количества путей на $10^9 + 7$

Формат входных данных

В первой строке ввода даны два целых числа n, m — количество перекрёстков и дорог в городе, где живет Денис ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq \min(2 \cdot 10^5, \frac{n(n-1)}{2})$).

Во второй строке даны три целых числа s, t и L ($1 \leq s, t \leq n, s \neq t, 1 \leq L \leq 2 \cdot 10^{14}$) — номер перекрёстка с домом Дениса, номер перекрёстка с университетом и время поездки Дениса.

В следующих m строках содержатся описания дорог между перекрёстками. Описание состоит из трёх целых чисел v, u и w ($1 \leq v, u \leq n, v \neq u, 1 \leq w \leq 10^9$) — номеров двух перекрёстков, которые связывает дорога, и времени, которое требуется для проезда по ней.

Все дороги двусторонние, между любой парой перекрёстков существует не более одной соединяющей их дороги. Никакая дорога не соединяет перекресток сам с собой. Гарантируется, что из перекрёстка s можно доехать до перекрёстка t .

Формат выходных данных

Выведите единственное целое число — количество маршрутов, по которым мог проехать Денис, взятое по модулю $10^9 + 7$.

Пример

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

Задача С. Турнир

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В лингвистической игре «Шляпа» могут принимать участие несколько пар игроков. Также в игре должен быть ведущий.

Учитель планирует организовать турнир по «Шляпе» в своем классе, в котором учатся n школьников, n нечетно. Для этого он хочет разбить школьников на пары, а тот, кто останется без пары — будет ведущим.

Пронумеруем школьников целыми числами от 1 до n . Для i -го школьника известен его уровень игры в «Шляпу» — целое число a_i . Уровень игры пары равен сумме уровней игры входящих в нее двух школьников.

Чтобы турнир был как можно более справедливым, учитель хочет, чтобы разность между максимальным и минимальным уровнями получившихся пар была как можно меньше. Помогите учителю выбрать ведущего и составить из оставшихся $n - 1$ школьников пары так, чтобы добиться поставленной цели.

Формат входных данных

Первая строка входных данных содержит целое число n — число учеников в классе ($3 \leq n \leq 5 \cdot 10^5$, гарантируется, что n нечетно).

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — минимальную возможную разность между максимальным и минимальным уровнем пар в турнире.

Пример

стандартный ввод	стандартный вывод
5 1 2 3 5 9	1

Задача D. Автомобильный туризм

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Маша и Миша решили отправиться в автомобильный поход. Они планируют ехать вдоль длинной прямой дороги, на которой расположены n заправок. Пронумеруем заправки вдоль дороги от 1 до n . Исходно машина расположена в городе, который находится в начале дороги, здесь же расположена первая заправка с номером 1.

Заправка с номером i располагается на расстоянии x_i от начала дороги. На этой заправке можно покупать бензин по цене p_i рублей за литр. На каждой заправке разрешается покупать только целое число литров бензина. Одного литра бензина хватает, чтобы проехать на расстояние 1 вдоль дороги. Вместимость бензобака машины составляет C литров, исходно он пуст.

У Миши и Маши есть бюджет в B рублей на покупку бензина. Как далеко вдоль дороги они смогут уехать, если об обратном пути они решили пока не думать?

Формат входных данных

Первая строка содержит три целых числа n , B и C — количество заправок, бюджет в рублях и вместимость бензобака в литрах, соответственно ($1 \leq n \leq 10^5$, $1 \leq B \leq 10^9$, $1 \leq C \leq 10^9$).

Каждая из следующих n строк содержит по два целых числа x_i и p_i — координату i -й заправки и стоимость литра бензина на ней, соответственно ($0 = x_1 < x_2 < \dots < x_n \leq 10^9$, $1 \leq p_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — максимальное расстояние, на которое можно уехать.

Пример

стандартный ввод	стандартный вывод
3 10 5 0 3 1 1 4 2	7

Замечание

В тесте из примера можно купить 1 литр бензина на первой заправке, проехать на расстояние 1, доехать до второй заправки, купить там 5 литров бензина, затем проехать на расстояние 3, доехав до третьей заправки. В этот момент в баке останется 2 литра бензина, после чего, купив еще 1 литр бензина, можно будет проехать расстояние 3, оказавшись на расстоянии 7 от начала дороги.

Задача E. кех

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим множество неотрицательных целых чисел A . Минимальное неотрицательное целое число, которое не встречается в этом множестве, часто возникает, например, в теории игр, и обозначается как $\text{mex}(A)$. Например, $\text{mex}(\{0, 1, 2, 4, 5, 9\}) = 3$.

Аня решила обобщить понятие mex . Рассмотрим целое положительное число k и множество целых неотрицательных чисел A . Обозначим как $\text{kex}(A, k)$ неотрицательное целое число, которое является k -м по возрастанию среди всех чисел, не входящих в A . Например, $\text{kex}(\{0, 1, 2, 4, 5, 9\}, 2) = 6$.

Требуется найти $\text{kex}(A, k_i)$ для заданного множества A и q значений k_i .

Формат входных данных

В первой строке входных данных дано два числа n и q ($1 \leq n, q \leq 10^5$) — количество элементов во множестве A и количество значений kex , которое надо найти.

Во второй строке в порядке возрастания даны n различных чисел, не превышающих 10^9 , — элементы множества A .

В третьей строке даны q чисел k_i ($1 \leq k_i \leq 10^9$).

Формат выходных данных

Выведите q значений: $\text{kex}(A, k_1), \text{kex}(A, k_2), \dots, \text{kex}(A, k_q)$.

Пример

стандартный ввод	стандартный вывод
4 10	0 3 4 5 8 9 10 11 13 14
1 2 6 7	
1 2 3 4 5 6 7 8 10 11	

Задача F. Среднее по Бобу

Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

На курсе по математической статистике Боб придумал новый способ вычисления среднего для массивов, содержащих нечетное число элементов.

Пока длина массива больше единицы, выполняется следующая операция: выбирается произвольный отрезок массива длины 3 с границами l и $r = l + 2$, вычисляется медиана этих трех элементов и затем эти элементы заменяются на один элемент, равный медиане.

Напомним, что медианой массива из трех элементов называется второй по возрастанию элемент данного массива. Например, медиана массива $[1, 5, 4]$ равна 4, а медиана массива $[2, 2, 2]$ равна 2.

Рассмотрим один из способов вычисления среднего по Бобу на массиве $[4, 1, 3, 2, 5]$. На первом шаге выбираем подотрезок с границами $l = 2, r = 4$. Так как 2 является медианой подмассива $[1, 3, 2]$, то массив преобразуется следующим образом: $[4, 1, 3, 2, 5] \rightarrow [4, 2, 5]$. На втором шаге единственный возможный отрезок длины 3 имеет границы $l = 1, r = 3$. Так как 4 является медианой подмассива $[4, 2, 5]$, то $[4, 2, 5] \rightarrow [4]$.

Боб заметил, что его способ вычисления среднего определен не вполне корректно: в зависимости от выбора отрезков результат вычисления может получиться разным. Чтобы исправить эту ситуацию, Боб решил выбирать отрезки для замены на медиану таким образом, чтобы единственное оставшееся в конце число было максимальным возможным. Именно это число Боб называет *средним по Бобу* для данного массива.

Для массива a из n элементов вам требуется ответить на q запросов, j -й из которых характеризуется границами L_j и R_j .

Ответом на i -й запрос является среднее по Бобу отрезка исходного массива $[a_{L_j}, a_{L_j+1}, \dots, a_{R_j}]$. Для всех запросов гарантируется, что длина отрезка запроса нечетна.

Формат входных данных

В первой строке содержится целое число n ($3 \leq n \leq 5 \cdot 10^4$) — размер массива.

Во второй строке содержатся n целых чисел a_i ($0 \leq a_i \leq 10^9$) — элементы массива.

В третьей строке содержится целое число q ($1 \leq q \leq 10^5$) — количество запросов.

В следующих q строках, содержатся целые числа L_j, R_j ($1 \leq L_j \leq R_j \leq n$) — границы подотрезка j -го запроса, гарантируется, что $R_j - L_j + 1$ не делится на 2.

Формат выходных данных

В j -й строке выходных данных выведите одно целое число — ответ на j -й запрос.

Пример

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

Задача G. Дорожка вокруг озера

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Леся работает ландшафтным дизайнером. Её задача — проложить вокруг озера в парке дорожку, используя как можно меньше строительных материалов. Дорожку можно строить только на клетках газона.

Парк представляет собой клетчатый прямоугольник размером $h \times w$, каждая клетка которого это либо земля, либо газон, либо озеро. Вам дана карта парка, на которой отмечены только клетки газона. Гарантируется, что:

- 1) множество клеток газона 4-связно, иначе говоря, можно добраться от любой клетки до любой другой, переходя от клетки к клетке только через общую сторону;
- 2) у множества клеток газона есть ровно одна внутренняя 4-связная область — озеро;
- 3) можно, начав в какой-то клетке газона, пройти по клеткам газона по циклу и вернуться в ту же клетку, переходя каждый раз в соседнюю клетку по стороне, так что озеро окажется внутри этого цикла. Будем говорить, что множество клеток, для которого выполнено это свойство, *4-окружает* озеро.

Требуется проложить дорожку по некоторым клеткам газона так, чтобы дорожка 4-окружала озеро, и клеток в дорожке было как можно меньше.

Предложите план прокладывания оптимальной дорожки.

Формат входных данных

В первой строке содержатся натуральные числа h и w ($3 \leq h, w \leq 1000$) — размеры парка.

Каждая из следующих h строк содержит w символов «#» и «.». Символ «#» означает клетку газона, а символ «.» означает клетку земли или клетку озера.

Формат выходных данных

Выведите h строк по w символов — оптимальный план прокладывания дорожки. Символ «#» означает клетку дорожки, а символ «.» означает клетку, не принадлежащую дорожке. Обратите внимание, что дорожку можно прокладывать только по клеткам газона.

Если существует несколько оптимальных планов проложить дорожку, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
4 6 ##### #.### #...# #####	#####. #...#. #...#. #####.

Задача Н. Одномерная игра

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Богдан играет в игру. Игра одномерная — в ней есть n платформ, расположенных на одной горизонтальной прямой. Между платформами можно перемещаться. i -я платформа это горизонтальный отрезок на прямой $[l_i, r_i]$. Все отрезки различны.

Будем говорить, что платформа j *вложена* в платформу i , если $l_i \leq l_j$ и $r_j \leq r_i$.

В начале игры Богдан выбирает платформу с которой он начнёт свой путь. Прыжки между платформами — занятие опасное, а Богдан старается держаться далеко от опасности, поэтому с платформы i он может прыгнуть на платформу j , только если платформа j вложена в платформу i , а также не существует другой платформы k , вложенной в платформу i , в которую вложена платформа j .

Для каждой платформы Богдану интересно сколько различных маршрутов он сможет пройти, начав с этой платформы и закончив на любой другой. Два маршрута считаются различными, если существует платформа, которая есть в одном маршруте, но которой нет в другом.

Помогите Богдану посчитать количество маршрутов с каждой платформы. Так как количество маршрутов может быть очень большим, необходимо вывести остаток от деления количества маршрутов на $10^9 + 7$.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество платформ.

Следующие n строк содержат описания платформ, i -я из них содержит два числа l_i и r_i ($1 \leq l_i \leq r_i \leq 10^9$) — концы отрезка i -й платформы.

Формат выходных данных

Выведите n чисел, i -е число должно быть равно количеству различных маршрутов, начинающихся в платформе i , взятое по модулю $10^9 + 7$.

Примеры

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

Задача I. Советские ясли

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

У Михаила Абрамовича, члена КПСС с 1961 года, доктора наук, научного атеиста, любящего мужа и отца, 10 лет назад родился внук по имени Максим. Но в отличие от своего дедушки Максим не увлекается науками, а вместо этого дни напролет играет в игры на телефоне.

Недавно Максим скачал себе новую игру на телефон — «Змейка 2022». Игровое поле для «Змейки 2022» представляет собой прямоугольную таблицу $n \times m$. Клеткой (r, c) будем называть клетку, которая находится на пересечении r -й строки и c -го столбца, пронумеруем строки сверху вниз от 1 до n , столбцы — слева направо от 1 до m . В каждой клетке поля находится яблоко, которое змейка съедает, когда ее голова оказывается в этой клетке. При этом игрок получает w_{ij} очков, когда змейка съедает яблоко, которое находится в клетке (i, j) .

В начале игры змейка имеет длину 1, ее голова появляется в клетке (a_r, a_c) и немедленно съедает расположенное там яблоко. Игра заканчивается, когда голова змейки оказывается в клетке (b_r, b_c) .

За один ход игрок перемещает голову змейки в любую из соседних клеток, которая пока не занята змейкой. Так как в каждой клетке находится яблоко, то змейка наращивает свою длину на 1 после каждого хода и множество клеток, занятых змейкой остается тем же, плюс к нему добавляется клетка, в которую переместилась её голова. Ход змейки описывается одним символом: «U» для перемещения вверх, из (r, c) в $(r - 1, c)$; «D» для перемещения вниз, из (r, c) в $(r + 1, c)$; «L» для перемещения влево, из (r, c) в $(r, c - 1)$; «R» для перемещения вправо, из (r, c) в $(r, c + 1)$.

Пусть W — суммарная стоимость яблок внутри поля, тогда считается, что игрок победил, если набрал строго больше $\frac{1}{2}W$ очков. При этом для упрощения игры гарантируется, что любое яблоко приносит строго меньше очков, чем суммарно приносят яблоки в клетках (a_r, a_c) и (b_r, b_c) .

Для Максима данная игра оказалась слишком сложной, он никак не может выиграть. Поэтому он обратился за помощью к своему дедушке. Михаил Абрамович же в ответ рассказал историю, как во время его молодости точно такую же задачу решил обычный советский детсадовец.

Вы выступаете в роли этого детсадовца, ваша задача заключается в том, чтобы для каждой конфигурации игрового поля из тестов предъявить выигрышную стратегию.

Формат входных данных

В первой строке дано число t — количество тестов во входных данных.

Каждый из тестов описывается в следующем формате. В первой строке указаны шесть чисел n, m, a_r, a_c, b_r, b_c — размеры поля, координаты начальной клетки и координаты конечной клетки ($2 \leq n, m \leq 5000, 1 \leq a_r, b_r \leq n, 1 \leq a_c, b_c \leq m$, начальная и конечная клетки различны).

Сумма $n \cdot m$ по всем тестам в одном наборе входных данных не превышает 10^6 .

В следующих n строках перечисляются стоимости яблок, находящихся в клетках поля, а именно, в i -й из этих строк содержатся целые числа $w_{i1}, w_{i2}, \dots, w_{im}$ ($1 \leq w_{ij} \leq 10^9$, гарантируется что для любых $1 \leq i \leq n$ и $1 \leq j \leq m$ выполняется неравенство $w_{ij} < w_{a_r a_c} + w_{b_r b_c}$).

Формат выходных данных

Для каждого тестового примера выведите строку, состоящую из символов «U», «D», «L», «R»: последовательность ходов змейки, выполняя которую её голова попадает из клетки (a_r, a_c) в клетку (b_r, b_c) , не посещает клетку, в которой уже находится змейка, и набирает больше, чем $\frac{1}{2}W$ очков.

Пример

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

Задача J. Строительство пирамиды

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Как известно, египетские пирамиды — одни из самых известных чудес света. Они были построены пришельцами в давние-давние времена. Пришелец Ар'глх, каждый день пролетая мимо Земли на работу и обратно, часто делает остановку, чтобы полюбоваться этими величественными сооружениями и найти вдохновение.

Видите ли, Ар'глх — архитектор и проектирует свою собственную пирамиду. К счастью, с новыми технологиями для постройки пирамиды ему не понадобятся ни огромные каменные блоки, ни труд множества египетских рабочих, но это все еще не значит, что его задача проста.

За последние столетия Ар'глх собрал n карбоновых треугольных граней, i -я из которых имеет стороны длиной a_i , b_i и c_i , соответственно. Он хочет выбрать четыре грани и собрать их в пирамиду в форме невырожденного тетраэдра. Разумеется, грани должны точно прилегать друг к другу без деформаций и пропусков и не должны выступать за тетраэдр. Объем тетраэдра должен быть строго положительным.

Помогите инопланетному архитектору определить, какие четыре грани он может выбрать из имеющегося у него набора, чтобы собрать из них пирамиду в форме тетраэдра.

Формат входных данных

В первой строке дано целое число n — количество треугольных граней в наличии у Ар'глха ($4 \leq n \leq 1500$)

В i -й из следующих n строк перечислены три целых числа a_i , b_i и c_i — длины сторон i -й грани ($1 \leq a_i, b_i, c_i \leq 10\,000$, гарантируется, что каждая грань является корректным треугольником, то есть $a_i + b_i > c_i$, $a_i + c_i > b_i$, $b_i + c_i > a_i$).

Формат выходных данных

Если Ар'глх сможет выбрать четыре грани, из которых можно собрать тетраэдр, выведите в первой строке слово «Yes». Во второй строке выведите четыре различных целых числа от 1 до n — номера граней, которые можно использовать для строительства. Если возможных ответов несколько, выведите любой из них.

Иначе выведите слово «No».

Примеры

стандартный ввод	стандартный вывод
7 3 3 3 3 4 5 4 5 6 3 4 3 3 3 5 6 5 7 3 5 5	Yes 2 4 5 7
5 2 3 4 3 4 5 4 5 6 5 6 7 6 7 8	No
4 2 3 4 3 4 2 3 2 4 4 2 3	No

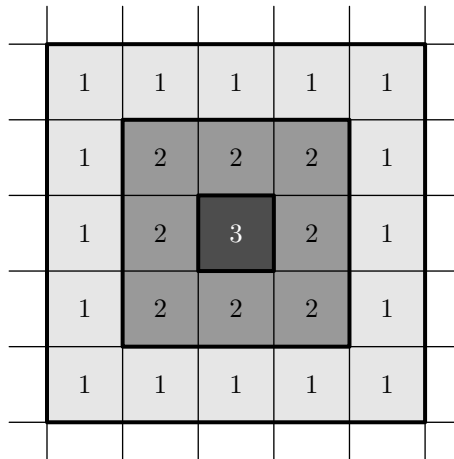
Задача К. Защита крепости

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Флатландцы строят крепость на клетчатой плоскости. Крепость будет иметь несколько контуров обороны, каждый из которых будет представлять собой прямоугольник со сторонами, проходящими по границам клеток.

Первый, внешний, контур обороны должен иметь форму прямоугольника размера $h \times w$, а каждый следующий контур должен находиться строго внутри всех предыдущих. Контуров обороны не должны иметь общих точек.

Уровнем защиты клетки назовем количество контуров обороны, внутри которых она лежит. Уровень защиты крепости равен сумме уровней защиты всех входящих в нее клеток. Например, уровень защиты крепости на рисунке равен $16 \cdot 1 + 8 \cdot 2 + 3 = 35$.



Флатландцы заинтересовались всеми возможными способами построить крепость. Для каждого способа они считают уровень защиты этой крепости, а затем вычисляют сумму этих значений по всем способам. Помогите им посчитать эту сумму. Ответ может быть довольно большим, выведите его остаток от деления на $10^9 + 7$.

Формат входных данных

На первой строке ввода находятся два целых числа h и w ($1 \leq h, w \leq 400$).

Формат выходных данных

Выведите одно число: сумму уровней защиты для всех способов построить крепость, взятую по модулю $10^9 + 7$.

Примеры

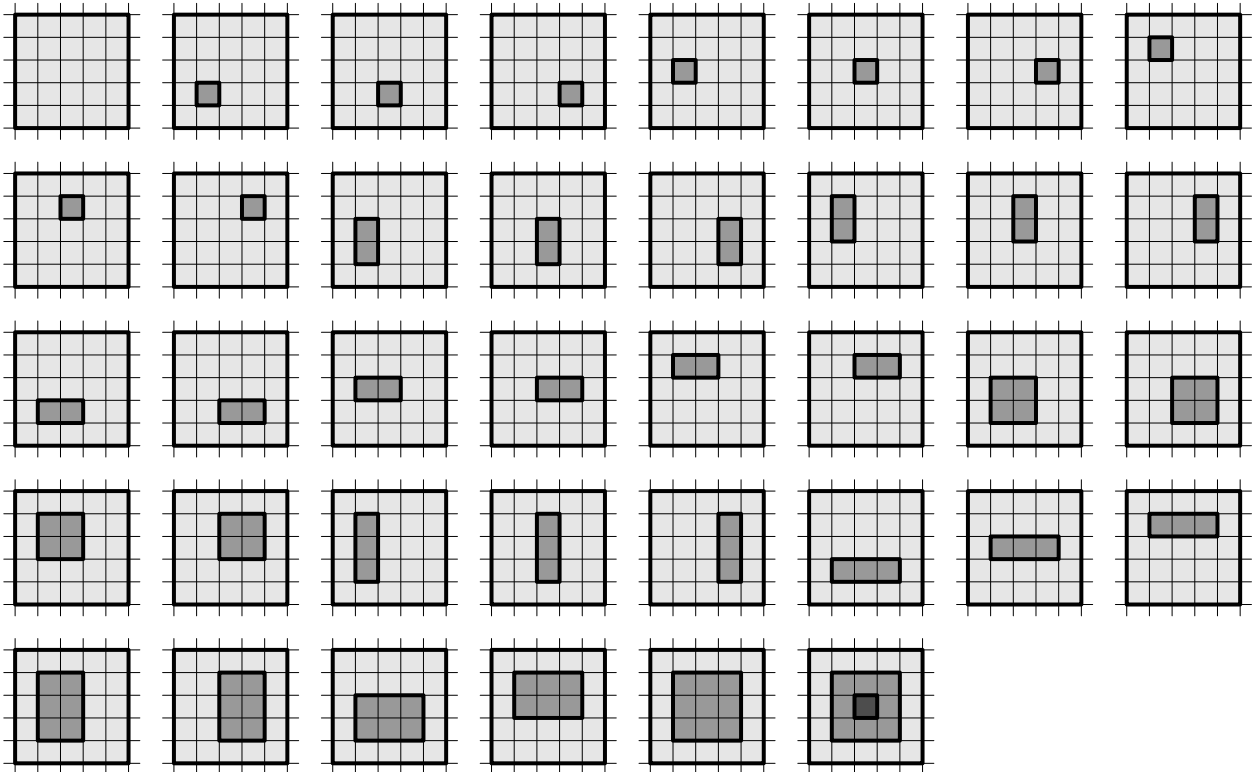
стандартный ввод	стандартный вывод
3 3	19
5 5	1060

Замечание

Все способы построить крепость в первом примере приведены на рисунке.



Все способы построить крепость во втором примере приведены на рисунке.



Задача L. Головоломка с проводами

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Это интерактивная задача. В процессе решения ваша программа должна взаимодействовать с программой жюри, используя стандартные потоки ввода и вывода.

Друзья играют в игру-головоломку, которая устроена следующим образом. Есть n одинаковых проводов, проложенных через непрозрачную трубу. Будем называть один из концов трубы левым, а другой — правым. С каждого из концов трубы игроки видят n проводов, которые помечены различными числами от 1 до n , при этом один и тот же провод с разных сторон может быть помечен разными номерами.

Цель игры — установить соответствие между концами проводов, то есть для каждого i от 1 до n определить, каким числом a_i помечен на левом конце трубы провод, помеченный числом i на правом конце трубы.

Для этого можно выполнять следующие запросы. Выберем число k ($1 \leq k \leq n$). После этого разобьём провод у правого конца трубы на k непустых групп и соединим все провода внутри одной группы. Затем с помощью специального устройства можно проверить для каждой пары проводов у их левого конца, соединены ли их правые концы. Таким образом, для левых концов проводов известно, помеченные какими числами слева провода попали в одну группу.

Требуется установить соответствие между левыми и правыми концами проводов, выполнив ровно три описанных запроса.

Для лучшего понимания действий, которые выполняются в задаче, настоятельно рекомендуем ознакомиться с примером, описанным ниже.

Протокол взаимодействия

Сначала ваша программа должна считать со стандартного потока ввода число n ($3 \leq n \leq 200$). Затем она должна выполнить три запроса.

Чтобы выполнить запрос, необходимо сначала вывести в стандартный поток вывода на отдельной строке число k — количество групп, а затем n целых чисел g_1, g_2, \dots, g_n . Каждое число должно быть в диапазоне от 1 до k , каждое число от 1 до k должно встречаться хотя бы один раз. число g_i задает номер группы, в которую определяется правый конец провода, помеченный числом i . Все правые концы проводов, попавшие в одну группу, соединяются.

В ответ на это программа жюри сообщит вашей программе, какие левые концы проводов соответствуют проводам, правые концы которых попали в одну группу. А именно, она выведет информацию о k группах. Информация о j -й группе начинается с числа s_j на отдельной строке, это число задает количество проводов в этой группе, а затем следует s_j чисел — номера, которыми помечены левые концы проводов, попавшие в эту группу. При этом порядок, в котором программа жюри перечисляет группы, может быть произвольным. В частности, он может отличаться от порядка, в котором группы были пронумерованы в запросе. Номера концов проводов в каждой группе также могут быть перечислены в произвольном порядке.

Выполнив три описанных запроса, ваша программа должна вывести n чисел a_1, a_2, \dots, a_n , где a_i — номер, которым помечен левый конец провода, правый конец которого помечен как i .

Пример

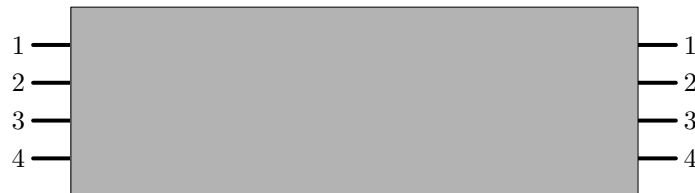
стандартный ввод	стандартный вывод
4	3
	1 1 3 2
2	
2 3	
1	
1	
1	
4	2
	1 1 1 2
1	
1	
3	
2 3 4	2
	2 1 2 2
1	
3	
3	
1 2 4	2 3 4 1

Обратите внимание, в примере ввод и вывод отформатированы с помощью пустых строк, чтобы было понятно, на какое сообщение какой ответ получен. В настоящем взаимодействии пустых строк не будет на вводе, и не надо их выводить.

Замечание

В приведенном примере события развиваются следующим образом.

В трубе находятся 4 провода.

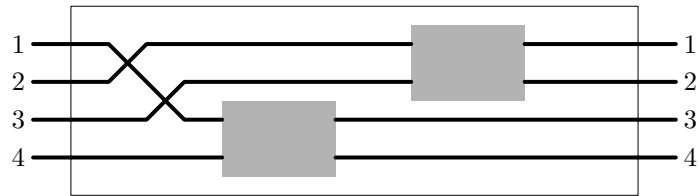


В первом запросе программа участника сделала три группы проводов, соединив провода с номерами правых концов 1 и 2, провод с номером правого конца 3 не соединен с другими, провод с номером правого конца 4 также не соединен с другими.

Программа жюри сообщает, что провода с номерами левого конца 2 и 3 в одной группе, провод с номером левого конца 1 в отдельной группе, провод с номером левого конца 4 также в отдельной группе.

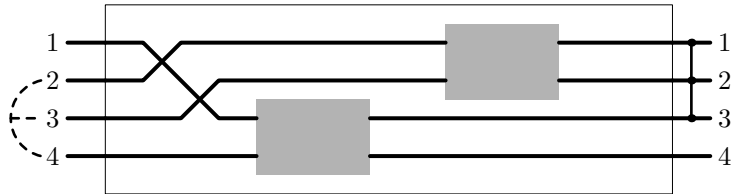


На основании этого программа участника понимает, что провода с номерами левых концов 2 и 3 соответствуют в некотором порядке проводам с номерами правых концов 1 и 2, а провода с номерами левых концов 1 и 4 — проводам с номерами правых концов 3 и 4.

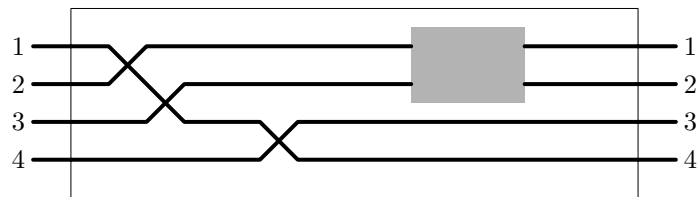


Вторым запросом программа участника создала две группы проводов и соединила правые концы проводов с номерами правых концов 1, 2 и 3, провод с номером правого конца 4 не соединен с другими.

Программа жюри сообщает, что объединены в группу оказались провода с номерами левых концов 2, 3 и 4.



Это позволило программе участника понять, что провод с правым концом 4 на левом конце помечен числом 1, а следовательно провод с правым концом 3 на левом конце помечен числом 4.



Третий запрос аналогично позволяет различить провода с правым концом 1 и 2, таким образом соответствие левых и правых концов полностью установлено.

