Задача A. Сеты в архиве исследователя

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Молодой исследователь обнаружил в старом университете пыльный ящик с загадочными карточками. На обороте ящика сохранилась надпись, гласящая, что эти карты использовались для тренировок внимания у студентов прошлых веков. Каждая карта содержит набор фигур, и по преданию, мастера логики соревновались друг с другом, пытаясь быстрее всех находить особые тройки карт — “сеты”.

Каждая карта характеризуется четырьмя признаками:

- числом фигур: `one`, `two` или `three`;
- цветом: `red`, `green` или `purple`;
- типом заливки: `open`, `solid` или `striped`;
- формой фигур: `oval`, `diamond`, `squiggle` (в единственном числе, если фигур одна) или `ovals`, `diamonds`, `squiggles` (во множественном числе, если фигур больше одной).

Три карты образуют *сет*, если по каждому из четырёх признаков выполняется одно из двух условий: либо все три карты имеют одинаковое значение признака, либо все три — различные.

Вам дан набор попарно различных карт из найденной коллекции. Требуется определить, сколько различных троек карт из этого набора образуют сеты. Тройки не обязаны быть непересекающимися.

Формат входных данных

Первая строка содержит целое число n ($3 \leqslant n \leqslant 81$) — количество карт. Следующие n строк описывают карты. Описание каждой карты состоит из четырех слов на английском языке — первое из них обозначает число фигур на карте, второе — цвет, третье — заливку, а четвертое — форму. При этом если фигура на карте одна, то ее тип указан в единственном числе (oval, diamond, squiggle), а если не одна, то во множественном (ovals, diamonds, squiggles).

Формат выходных данных

Выведите одно число — количество троек карт, образующих сеты.

Примеры

стандартный ввод	стандартный вывод
3 one red solid oval two green solid ovals three purple solid ovals	1
3 one red open oval two green open diamonds three purple striped squiggles	0

Задача B. Аномалия степени сети

Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Исследовательская группа изучает небольшую пиринговую коммуникационную сеть, состоящую из N устройств. Каждое устройство может устанавливать двустороннюю связь с любым количеством других устройств, и эти коммуникационные связи должны образовывать неориентированный простой (без кратных ребер и петель) граф на N вершинах.

Во время диагностического сканирования система измерений зафиксировала степень каждого устройства—количество других устройств, с которыми оно напрямую общается. Однако из-за сбоя в журналировании к списку было добавлено ровно одно лишнее значение степени. В результате система выдала $N+1$ целых числа вместо N, появляющихся в порядке, в котором они были прочитаны.

Ваша задача состоит в том, чтобы определить, какие записи в списке могут быть ошибочным лишним значением. Чтобы индекс был допустимым кандидатом, удаление значения в этой позиции должно оставить последовательность из N неотрицательных целых чисел, которые могут представлять последовательность степеней некоторого простого неориентированного графа на N помеченных вершинах.

Формат входных данных

Первая строка содержит целое число N ($2 \leqslant N \leqslant 500$). Каждая из следующих $N+1$ строк содержит одно целое число, представляющее одно зарегистрированное значение степени, в порядке, выданном системой измерений. Все значения являются неотрицательными целыми числами.

Формат выходных данных

В первой строке выведите целое число K, обозначающее, сколько записей могут быть ложным лишним значением. На каждой из следующих K строк выведите индекс от 1 до $N+1$, удаление которого дает допустимую последовательность степеней. Индексы должны быть выведены в порядке возрастания.

Пример

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

Замечание

После удаления первого элемента оставшаяся последовательность $2, 2, 1, 3$ соответствует допустимому простому неориентированному графу. Одно из таких воплощений использует четыре помеченные вершины A, B, C и D, где A соединен с B, C и D; B и C также смежны; а D не имеет других рёбер. Удаление четвертого элемента (другого 1) дает тот же мультимножество степеней и также допустимо. Удаление последнего элемента (3) оставляет $1, 2, 2, 1$, что также является графическим. Удаление любого из 2 оставляет последовательность с нечётной общей суммой, что делает допустимый граф невозможным; следовательно, эти индексы не включены в вывод.

Задача C. Игра жизни на круге

Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Денис и Демид придумали новую вариацию “Игры Жизни”.

Существует N ячеек, расположенных в круге и пронумерованных от 0 до $N-1$. Для каждой ячейки i $(0 \leqslant i \leqslant N-2)$ ячейка i соседствует с ячейкой $i+1$, а ячейка $N-1$ соседствует с ячейкой 0. Таким образом, каждая ячейка имеет ровно двух соседей.

Каждая ячейка находится в одном из двух состояний: `живой` или `мертвый`.

Начальная конфигурация в момент времени 0 выбирается Денисом и Демидом. Для каждого времени $t > 0$ состояние каждой ячейки обновляется следующим образом:

- Рассмотрим ячейку i и её две соседние ячейки.
- Если как минимум две из этих трех ячеек `живые` в момент времени $t-1$, то ячейка i становится `живой` в момент времени t.
- Если как минимум две из этих трех ячеек `мертвые` в момент времени $t-1$, то ячейка i становится `мертвой` в момент времени t.

Вам дана строка s длиной N. Каждый символ является либо `'0'`, либо `'1'`, либо `'?'`. Символы `'0'` и `'1'` указывают на начальное состояние соответствующей ячейки, в то время как `'?'` означает, что состояние не определено.

Пусть T — заданный момент времени. Ваша задача — подсчитать количество способов заменить все символы `'?'` на `'0'` или `'1'` так, чтобы как минимум K ячеек находились в состоянии `живой` в момент времени T.

Формат входных данных

Входные данные состоит из строки s $(3 \leqslant |s| \leqslant 50)$, за которой следует целое число T $(0 \leqslant T \leqslant 1000)$, а затем целое число K $(0 \leqslant K \leqslant |s|)$. Каждый символ является `'0'`, `'1'`, или `'?'`.

Формат выходных данных

Выведите количество допустимых назначений для неопределенных ячеек.

Примеры

стандартный ввод	стандартный вывод
0?1 1 1	1
????????? 0 1	511

Замечание

В первом примере существует два возможных начальных состояния: `"001"` и `"011"`. После одного временного шага они эволюционируют в `"000"` и `"111"`, соответственно. Только вторая конфигурация имеет хотя бы одну `живую` ячейку в момент времени $T = 1$, поэтому ответ равен 1.

Задача D. Мутационная последовательность

Ограничение по времени: 0.3 секунд
Ограничение по памяти: 256 мегабайт

На недавно открытой экзопланете PO-2025 ДНК обитателей представлено строкой, состоящей только из букв `A`, `B` и `C`. Жизнь на PO-2025 развивалась быстрее, чем на Земле, что позволяет даже новорожденным решать программные задачи.

Вместо обычных мутаций на PO-2025 происходят «мультации». Мультация заменяет все вхождения одной буквы в строке ДНК на новую строку, содержащую от 1 до 3 букв (также только `A`, `B` или `C`). Повторяющиеся мультации могут быстро увеличить длину строки ДНК.

Ваша задача — найти кратчайшую последовательность мультаций, которая преобразует строку S в строку T. Гарантируется, что существует решение, использующее не более 3 мультаций.

Формат входных данных

Первая строка содержит строку S. Вторая строка содержит строку T. Обе строки имеют длину не более 10 и состоят только из букв `A`, `B` и `C`.

Формат выходных данных

Выведите одну строку для каждой мультации в порядке их применения. Каждая строка должна содержать две строки: букву, которая изменяется, и строку, на которую она заменяется.

Если существует несколько оптимальных последовательностей, выведите любую из них.

Примеры

стандартный ввод	стандартный вывод
ABA CBC	A A A A A C
BC CACCAB	B A C CAB A CA

Задача E. Построение дерева кабелями

Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Группа инженеров собирает прецизионную коммуникационную структуру. Дизайн предполагает связанную сеть из N устройств, расположенных в виде дерева с N вершинами и $N-1$ ребрами единичной длины. Каждое ребро описывается двумя целыми числами A и B, что указывает на необходимость прямого соединения соответствующих устройств.

Инженеры могут закупать коммуникационные кабели у поставщика. Кабель длиной L стоит ровно L^2. Однако поставщик накладывает строгие правила: кабели нельзя резать, удлинять или соединять; единственной разрешенной операцией является пайка, при которой конец одного кабеля прикрепляется к какой-то точке строго внутри другого кабеля. Пайка от конца до конца запрещена, но пайка нескольких концов к одной внутренней точке разрешена.

С помощью пайки инженеры хотят собрать точное дерево, указанное в дизайне. Они могут использовать любое количество купленных кабелей, но должны воспроизвести все ребра дерева, используя сегменты единичной длины, образованные через их операции пайки. Цель состоит в том, чтобы минимизировать общую стоимость закупленных кабелей.

Учитывая описание дерева, определите наименьшую возможную общую стоимость, необходимую для его сборки.

Формат входных данных

Первая строка содержит одно целое число N ($1 \leqslant N \leqslant 50\,000$). Каждая из следующих $N-1$ строк содержит два целых числа, описывающих ребро между вершинами A и B, где $1 \leqslant A, B \leqslant N$ и $A \neq B$.

Формат выходных данных

Выведите одно целое число, равное минимальной общей стоимости, необходимой для построения дерева.

Пример

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

Замечание

В этом примере можно купить кабель длиной 2 и три кабеля длиной 1. Пайкой более длинного кабеля соответствующим образом можно собрать все пять ребер единичной длины звездообразного дерева. Таким образом, общая стоимость составляет $2^2 + 1^2 + 1^2 + 1^2 = 7$.

Задача F. Восстановите холст Севы

Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Сева только что установил новую программу для рисования, которая поддерживает K различных цветов, пронумерованных от 1 до K. Все рисование выполняется на холсте размером $N \times N$. Изначально все ячейки белые (цвет 1). Ячейка в верхнем левом углу имеет координаты $(0, 0)$, при этом первая координата x увеличивается вниз по строкам, а вторая координата y увеличивается вправо по столбцам.

Сева любит создавать прямоугольные шахматные узоры, используя команду `PAINT c x1 y1 x2 y2`, где c — это цвет, а (x_1, y_1) и (x_2, y_2) — координаты верхнего левого и нижнего правого углов прямоугольника. Ячейка в верхнем левом углу прямоугольника окрашивается в цвет c, а остальные ячейки следуют шахматному узору. Ячейки, которые не были закрашены, сохраняют свой предыдущий цвет.

Кроме того, Сева может сохранить текущее состояние холста, используя команду `SAVE`, и восстановить ранее сохраненное состояние с помощью `LOAD x`, где x — это порядковый номер сохранения.

Программа выдала ошибку, и рисование Севы было потеряно. К счастью, остался журнал всех команд. Ваша задача — восстановить финальное изображение.

Формат входных данных

Первая строка содержит три целых числа N, K и M ($1 \leqslant N \leqslant 1000$, $2 \leqslant K \leqslant 100\,000$, $1 \leqslant M \leqslant 100\,000$), представляющих размер холста, количество цветов и количество команд соответственно. Каждая из следующих M строк содержит одну команду, либо `PAINT c x1 y1 x2 y2`, `SAVE`, или `LOAD x`. Гарантируется, что все команды являются корректными.

Формат выходных данных

Выведите N строк, каждая из которых содержит N целых чисел. Каждая строка соответствует строке холста, а каждое целое число представляет цвет ячейки в этой строке.

Примеры

стандартный ввод	стандартный вывод
4 3 2 PAINT 2 0 0 3 3 PAINT 3 0 3 3 3	2 1 2 3 1 2 1 2 2 1 2 3 1 2 1 2
3 4 7 PAINT 2 0 0 1 1 SAVE PAINT 3 1 1 2 2 SAVE PAINT 4 0 2 0 2 LOAD 2 PAINT 4 2 0 2 0	2 1 1 1 3 1 4 1 3

Задача G. Большой сюрприз

Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Тетрис-Сити — самый населённый и современный город в Неверленде, построенный на ровной местности с бесконечными скоплениями высотных зданий, расположенных так близко друг к другу, что они напоминают игру в Тетрис. Здания выглядят как параллельные оси коробки, построенные на земле, и они разрозненные (даже не касаются друг друга).

Большой сюрприз, объявленный мэром, будет специальной службой доставки с использованием дронов. Дроны, используемые в этой службе, представляют собой поколение квадрокоптеров, которые могут физически перемещаться только в одном из направлений x, y и z.

Таким образом, расстояние, пройденное дроном, — это сумма расстояний, пройденных им по каждой оси. Молодой мэр теперь приказал сделать дронов умными, оснастив их программным обеспечением, которое вычисляет кратчайший путь от любого источника к любой цели, избегая зданий. Ваша задача — разработать это программное обеспечение.

Формат входных данных

Первая строка входных данных содержит целое число n ($0 \leqslant n \leqslant 100$), указывающее количество зданий в Тетрис-Сити. Каждая из следующих n строк содержит 5 целых числа, разделённых пробелами: x, y, x', y' и h, указывающих здание: координаты (x, y) и (x', y') соответственно указывают на юго-западный угол и северо-восточный угол здания, а h определяет его высоту. Гарантируется, что объем здания не равен нулю. Исходные и конечные точки указаны в конце входных данных в двух отдельных строках; каждая из них содержит координаты x, y и z.

Все числа во входных данных — неотрицательные целые числа, не превышающие 100 000. Гарантируется, что исходная и конечная точки находятся вне зданий (они могут находиться на границе зданий). Кратчайший путь может касаться зданий, и предполагается, что дрон выглядит как точка.

Формат выходных данных

В выходных данных выведите длину кратчайшего пути от источника до назначения, избегая зданий.

Пример

стандартный ввод	стандартный вывод
1 1 1 11 21 40 5 0 5 6 23 8	35

Задача H. Контрольные ворота паспортного контроля

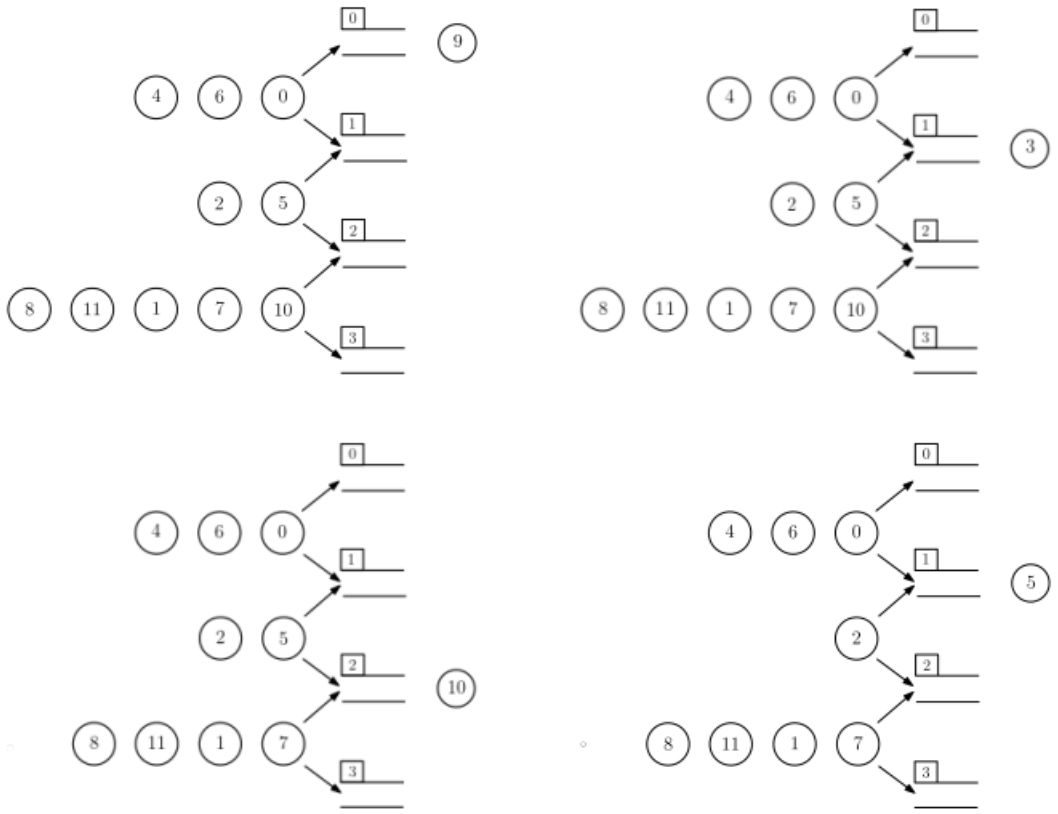
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Эверленд и Неверленд — две соседние страны, и каждый год огромное количество эверлендских туристов посещает Неверленд. Правительства хотят проанализировать процесс паспортного контроля на границе Неверленда и Эверленда и нуждаются в вашей помощи!

На границе туристы стоят в q очередях для проверки паспортов, и есть $q+1$ контрольных ворот. Если мы пронумеруем ворота от 0 до q, а очереди от 0 до $q-1$, то туристы, стоящие в очереди i, могут проходить только через ворота i или $i+1$.

Когда открывается ворота i, если одна из очередей i и $i-1$ пуста или не существует, первый турист другой очереди проходит через ворота. Если обе очереди i и $i-1$ не пусты, то старший турист из тех, кто стоит первыми в каждой из двух очередей, проходит через ворота. Предполагается, что ни одни два ворота не открываются одновременно.

У нас есть изображение n туристов, стоящих в очередях и ожидающих открытия ворот. Также у нас есть другое изображение, которое было сделано немного позже, на котором некоторые из туристов с первого изображения прошли через ворота. Туристы на изображениях пронумерованы от 0 до $n-1$ в порядке их возраста, так что человек номер 0 является самым молодым, а человек номер $n-1$ — самым старшим. На рисунке ниже показаны первые четыре конфигурации туристов в первом примере. Вам нужно найти любую допустимую последовательность открытия ворот, которая могла произойти между моментами, когда были сделаны два изображения, или заявить, что это невозможно. Ворота могут открываться последовательно несколько раз.



Формат входных данных

Первая строка входных данных содержит два целых числа q ($1 \leqslant q \leqslant 100\,000$) — количество очередей, и n ($0 \leqslant n \leqslant 100\,000$) — количество туристов на первом изображении. i-я строка из следующих q строк описывает очередь $i-1$ на первом изображении. Каждое описание очереди начинается с числа k ($0 \leqslant k \leqslant n$), которое показывает размер очереди, за которым следуют k

целых чисел, указывающих номера туристов в этой очереди, от конца к началу. Номера туристов уникальные неотрицательные целые числа, меньшие n. В следующих q строках появляется описание второго изображения в том же формате.

Формат выходных данных

Если нет допустимой последовательности, выведите `Impossible`. Если есть допустимые последовательности, выведите любую из них в следующем формате. Выведите длину последовательности в первой строке и саму последовательность во второй строке. В вашей последовательности каждый раз, когда открываются любые ворота, должен быть хотя бы один турист, ожидающий их.

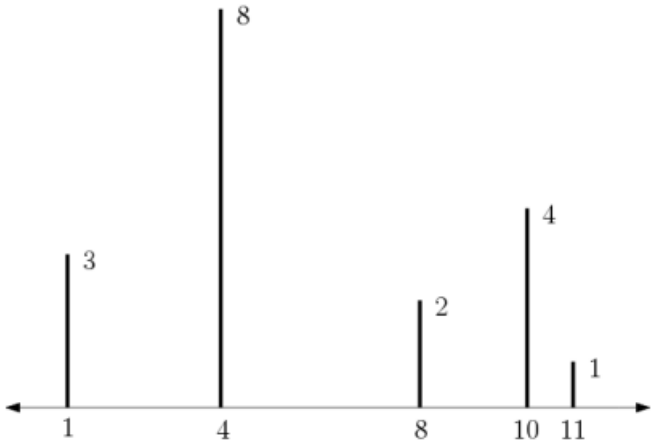
Примеры

стандартный ввод	стандартный вывод
3 12 4 4 6 0 9 3 2 5 3 5 8 11 1 7 10 3 4 6 0 1 2 2 8 11	6 0 1 2 1 2 3
3 3 1 2 1 0 1 1 1 2 0 1 1	Impossible
1 2 2 0 1 2 1 0	Impossible
1 2 2 0 1 2 0 1	0

Задача I. Жадный термит

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Существует n деревянных стержней, вертикально расположенных над горизонтальной линией. Стержни пронумерованы от 1 до n слева направо. Каждый стержень i ($1 \leqslant i \leqslant n$) расположен в позиции x_i и имеет высоту h_i.



Термит хочет поедать все стержни по одному. Он начинает есть с произвольного стержня s ($1 \leqslant s \leqslant n$). Затем, после поедания стержня i, термит выбирает следующий стержень для поедания на основе следующего метода. Среди оставшихся стержней j выбирается тот, у которого максимальное значение $h_j - |x_i - x_j|$. Если есть несколько кандидатов, выбирается тот, у которого минимальное значение $|x_i - x_j|$. Если все еще есть несколько кандидатов, выбирается самый левый стержень.

Ваша задача — вычислить общее (горизонтальное) расстояние, пройденное термитом, чтобы съесть все стержни.

Формат входных данных

Первая строка входных данных содержит два целых числа, разделенных пробелом, n, количество стержней, и s, номер стартового стержня ($1 \leqslant s \leqslant n \leqslant 100\,000$). Стержни описываются в следующих n строках. В строке $1 + i$ ($1 \leqslant i \leqslant n$) указаны i-й стержень с двумя целыми числами, разделенными пробелом, x_i ($|x_i| \leqslant 10^9$) и h_i ($1 \leqslant h_i \leqslant 10^9$). Кроме того, для каждого i ($1 \leqslant i \leqslant n - 1$) выполняется условие $x_i < x_{i+1}$.

Формат выходных данных

Вы должны вывести одно целое число, обозначающее общее расстояние, пройденное термитом.

Пример

стандартный ввод	стандартный вывод
5 3 1 3 4 8 8 2 10 4 11 1	17

Задача J. План Б

Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

В Кризисной Земле время от времени происходят кризисы, которые серьезно угрожают безопасности страны. В недавнем кризисе в нескольких городах страны произошла серия гражданских протестов. Протест начался в одном городе и быстро распространился на другие города. Чтобы предотвратить это в будущем, правительство в качестве Плана Б — после отключения интернета по всей стране — решило быстро отправить армию, чтобы окружить город, из которого начинается протест. Город считается окруженным, если в каждом из его соседних городов находятся армейцы. У правительства есть военные базы в b различных городах, каждая из которых имеет много армейцев, которые могут быть отправлены во все города. Правительство знает, что его армия не может пройти через город, из которого начинается протест, так как они могут быть убиты. Учитывая это, может случиться так, что некоторые города невозможно окружить армией. Эти города называются критическими. Предполагается, что если в городе есть военная база, то этот город не является критическим. Теперь правительство хочет узнать, есть ли критические города в стране или нет. Помогите правительству найти ответ.

О, мы забыли объяснить структуру Кризисной Земли! Чтобы разрешить этот кризис, следует упомянуть, что Кризисная Земля состоит из n городов, пронумерованных от 1 до n. Города соединены m дорогами, которые могут использоваться в обоих направлениях. Два города являются соседями, если между ними есть дорога. Гарантируется, что дорожная сеть Кризисной Земли связна.

Формат входных данных

Первая строка входных данных содержит три положительных целых числа n, m и b, обозначающих количество городов, дорог и военных баз соответственно ($1 \leqslant b \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 200\,000$). Каждая из следующих m строк содержит два числа v_i и u_i, обозначающих дорогу между городами v_i и u_i. Последняя строка состоит из b целых чисел, представляющих города, имеющие военную базу.

Формат выходных данных

Выходные данные состоят из двух строк. Первая строка содержит количество критических городов. Вторая строка содержит критические города в порядке возрастания.

Пример

стандартный ввод	стандартный вывод
7 8 3 1 2 1 3 1 4 2 5 2 6 5 6 3 4 3 7 4 5 6	1 3

Задача K. Лодка

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Река отделяет Верхний Тагил от Нижнего Тагила. Для транспортировки людей между этими двумя городами предусмотрена двухместная лодка (лодка, которая может перевозить не более двух человек) с определённой грузоподъемностью. Этой лодкой должен управлять как минимум один человек, т.е. она не может двигаться через реку без пассажиров.

Национальный фестиваль Тагила запланирован в Верхнем Тагиле. Все жители Нижнего Тагила хотят участвовать в этом празднике и нуждаются в том, чтобы как можно быстрее добраться до Верхнего Тагила.

Ваша задача — помочь им добраться до Верхнего Тагила с минимальным количеством поездок на лодке через реку.

Формат входных данных

Первая строка входных данных содержит два целых числа n и w, где n — количество жителей Нижнего Тагила ($1 \leqslant n \leqslant 1000$), а w — максимальный вес, который может нести лодка ($1 \leqslant w \leqslant 10^6$). Следующая строка содержит n целых чисел, разделённых пробелами, описывающих веса жителей Нижнего Тагила. Все веса — положительные целые числа, не превышающие 10^6.

Формат выходных данных

Если невозможно перевезти всех жителей Нижнего Тагила, выведите в выходных данных одну строку, содержащую “-1”. В противном случае выведите минимальное количество раз, которое лодка должна совершить поездки между Нижним Тагилом и Верхним Тагилом (в обе стороны), чтобы перевезти всех жителей Нижнего Тагила в Верхний Тагил.

Примеры

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

Задача L. ПВО

Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Мы хотим нанести стратегический ракетный удар по критическим ресурсам противника. Противник развернул системы противовоздушной обороны для защиты этих ресурсов. Однако их система обороны имеет определенные уязвимости, и ваша задача — эффективно их использовать.

Каждая система противовоздушной обороны может защищать все стратегические ресурсы и системы противовоздушной обороны в пределах своего радиуса действия, но не может защищать саму себя. Из-за технических ограничений каждый критический ресурс или система противовоздушной обороны защищена не более чем одной другой системой противовоздушной обороны.

Ракеты могут быть использованы для уничтожения либо незащищенного стратегического ресурса, либо системы противовоздушной обороны.

Ресурс считается незащищенным, если его не защищает ни одна активная система противовоздушной обороны. Когда система противовоздушной обороны уничтожена, она больше не может защищать никакие ресурсы или другие системы противовоздушной обороны. Ваша цель — максимизировать количество уничтоженных стратегических ресурсов.

Формат входных данных

Первая строка содержит три целых числа m (количество ракет), n (количество стратегических ресурсов) и d (количество систем противовоздушной обороны), где ($0 \leqslant m, n, d \leqslant 5000$).

Следующие n строк содержат по два целых числа x_i и y_i ($0 \leqslant x_i, y_i \leqslant 10^9$), координаты i-го стратегического ресурса.

Следующие d строк каждая содержат по три целых числа x_j, y_j и r_j ($0 \leqslant x_j, y_j \leqslant 10^9$; $0 \leqslant r_j \leqslant 10^9$), координаты j-й системы противовоздушной обороны и ее радиус защиты.

Формат выходных данных

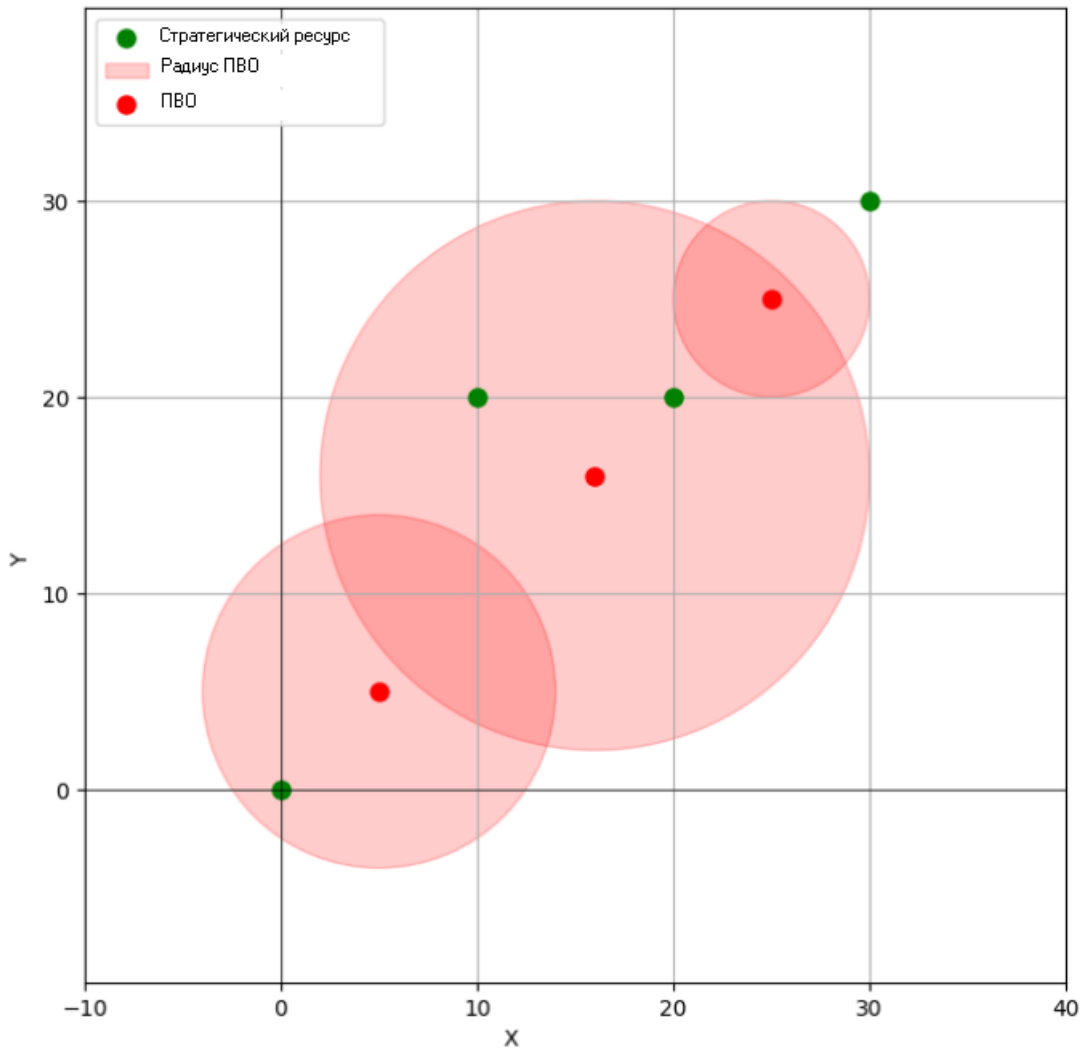
Выведите максимальное количество стратегических ресурсов, которые могут быть уничтожены.

Пример

стандартный ввод	стандартный вывод
3 4 3 0 0 10 20 20 20 30 30 5 5 9 16 16 14 25 25 5	2

Замечание

Пример набора входных данных изображен на рисунке ниже:


Стратегический ресурс
Радиус ПВО
ПВО
30
20
10
0
Y
−10
0
10
20
30
40
X

Задача M. Три этапа теста

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Автономное исследовательское учреждение планирует трехэтапный тест, используя свой парк экспериментальных дронов. Доступно N дронов, пронумерованных от 1 до N. Дрон i требует A_i миллисекунд для завершения одного 100-метрового этапа полета, а его задержка стабилизации составляет B_i миллисекунд, что представляет собой время, необходимое для получения контроля после получения навигационного ядра или для безопасной передачи его другому дрону.

Тестовый маршрут составляет ровно 300 метров и его нужно преодолеть с помощью ровно трех различных дронов, каждый из которых покрывает один 100-метровый этап. Навигационное ядро должно быть передано от первого дрона ко второму, а затем от второго к третьему. Если дрон i пролетает первый этап, дрон j второй, а дрон k третий, то тест проходит следующим образом. Дрон i завершает свой 100-метровый этап за A_i миллисекунд. Затем он передает навигационное ядро дрону j, что требует $\max(B_i, B_j)$ миллисекунд. Дрон j пролетает второй этап за A_j миллисекунд и передает навигационное ядро дрону k, что занимает $\max(B_j, B_k)$ миллисекунд. Наконец, дрон k пролетает последний этап за A_k миллисекунд. Общая продолжительность теста составляет

$$A_i + \max(B_i, B_j) + A_j + \max(B_j, B_k) + A_k.$$

В качестве главного оператора вы должны выбрать три дрона и определить их порядок так, чтобы общее время теста было как можно меньше.

Формат входных данных

Первая строка содержит N ($3 \leqslant N \leqslant 200000$). Каждая из следующих N строк содержит два целых числа A_i и B_i ($1 \leqslant A_i, B_i \leqslant 10^8$), представляющих время полета и задержку стабилизации дрона i.

Формат выходных данных

Выведите одно целое число: минимально возможную общую продолжительность теста, измеряемую в миллисекундах.

Примеры

стандартный ввод	стандартный вывод
4 1070 90 1080 70 1050 60 1020 100	3320
5 1000 28 1000 14 1000 21 1000 20 1000 14	3034

Замечание

В этом примере одним из оптимальных назначений является позволить дрону 4 пролететь первый этап, дрону 3 второй, а дрону 2 третий. Первый полет занимает 1020 миллисекунд, первая передача требует $\max(100, 60) = 100$ миллисекунд, второй полет занимает 1050 миллисекунд, вторая передача требует $\max(60, 70) = 70$ миллисекунд, а финальный полет занимает 1080 миллисекунд. Общее время составляет $1020 + 100 + 1050 + 70 + 1080 = 3320$ миллисекунд, что является минимально возможным.

Задача N. Лингвистическая дуэль

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вы проникаете в высокозащищенное ядро инопланетного сервера, и единственный способ пройти мимо последней защиты — это переиграть его охранный ИИ в лингвистическом поединке. ИИ работает с использованием секретного циклического ранжирования всех пятибуквенных слов, состоящих из заглавных букв от A до J (включительно). Таких слов ровно 10^5.

Ранжирование представляет собой строгий тотальный порядок: для любых двух различных слов одно превосходит другое. Единственное исключение — слово с самым низким рангом превосходит слово с самым высоким рангом, образуя единый цикл. Вы не знаете этого ранжирования.

Поединок длится 201 ход. На ходе 0 вы выводите слово W_0. Затем, для каждого хода $i = 1$ до 200:

- Если i нечетное, ИИ выводит слово W_i, которое превосходит W_{i-1}, выбирая его равномерно случайным образом среди всех слов, которые превосходят W_{i-1}.
- Если i четное, вы выводите слово W_i, и вы получаете очко, если W_i превосходит W_{i-1}.
- Если $W_i = W_{i-1}$, очко не начисляется.

Вы делаете первый и последний ходы (ходы 0 и 200), так что вы играете 101 слово, а ИИ — 100. Ваша цель — набрать как минимум 50 очков за ваши 100 ответных ходов (ходы 2, 4, ..., 200).

Протокол взаимодействия

Это интерактивная задача. Ваша программа должна читать из стандартного ввода и писать в стандартный вывод.

Сначала прочитайте одно целое число T ($1 \leqslant T \leqslant 50$). Затем, для каждого из T тестовых случаев:

Выведите пятибуквенное слово W_0, состоящее только из заглавных букв от A до J. Затем повторите следующее 100 раз: прочитайте пятибуквенное слово W_{2j-1} (гарантированно состоящее только из букв A–J) и выведите ваше следующее пятибуквенное слово W_{2j} (также используя только буквы A–J).

После вывода W_{200} тестовый случай заканчивается. Если остаются другие тестовые случаи, немедленно начните следующий, выведя его W_0.

Если ваша программа выводит недопустимое слово (неправильной длины или содержащее любую букву вне диапазона A–J), судья отвечает `-1` и завершает работу. При чтении `-1` ваша программа должна немедленно завершиться.

Судья использует фиксированное случайный сид для каждого тестового набора, поэтому детерминированные решения будут иметь одинаковые взаимодействия при разных запусках. Всегда очищайте стандартный вывод после каждого вывода.

Гарантируется, что используется не более 20 тестов (кроме примера).

Пример

стандартный ввод	стандартный вывод
1 AAAAB BDFED ...	 AAAAA BEAAA AAAAB ...

Замечание

Пример в условии не содержит всех 201 раунда, а только 5, чтобы ознакомиться с процессом. В тестовой системе вы будете играть все 201 раунд, и на первом тесте вы получите OK, если наберете как минимум 10 очков.