

Задача А. Объединение двусвязных списков

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Двусвязный список — одна из фундаментальных структур данных. Двусвязный список представляет собой последовательность элементов, в каждом из которых содержится информация о предыдущем и следующем элементах списка. В этой задаче все списки имеют линейную структуру: то есть у каждого элемента списка кроме первого есть ровно один предыдущий элемент, у каждого элемента списка кроме последнего есть ровно один следующий элемент, список не замкнут в цикл.

В этой задаче вам даны n ячеек памяти, образующие один или более двусвязных списков. Каждая ячейка памяти содержит информацию об элементе одного из списков. Ячейки памяти пронумерованы от 1 до n .

Для каждой ячейки памяти i известен номер ячейки, в которой содержится предыдущий элемент списка (величина l_i) и номер ячейки, в которой содержится следующий элемент списка (величина r_i). Если ячейка памяти i содержит информацию о таком элементе, у которого нет предыдущего элемента в списке, то $l_i = 0$. Аналогично, если ячейка памяти i содержит информацию о таком элементе, у которого нет следующего элемента в списке, то $r_i = 0$.



На рисунке изображены три списка.

Например, для рисунка выше значения l и r следующие: $l_1 = 4, r_1 = 7; l_2 = 5, r_2 = 0; l_3 = 0, r_3 = 0; l_4 = 6, r_4 = 1; l_5 = 0, r_5 = 2; l_6 = 0, r_6 = 4; l_7 = 1, r_7 = 0$.

Объедините все заданные списки в один, подсоединив их друг к другу в любом порядке. В частности, если входные данные уже содержат один список, то каких-либо действий производить не надо. Выведите результирующий список в виде пар значений l_i, r_i .

Каких-либо других действий, кроме как подсоединений начала одного списка к концу другого, совершать нельзя.

Формат входных данных

Первая строка входных данных содержит целое число n ($1 \leq n \leq 100$) — количество ячеек памяти, в которых расположены двусвязные списки.

Следующие n строк содержат по два целых числа l_i, r_i ($0 \leq l_i, r_i \leq n$) — номера ячеек памяти предыдущего и следующего элемента списка для ячейки памяти i . Величина $l_i = 0$, если у элемента ячейки памяти i нет предыдущего элемента в его списке. Величина $r_i = 0$, если у элемента ячейки памяти i нет следующего в его списке.

Гарантируется, что входные данные содержат описание одного или более двусвязных списков. Все списки имеют линейную структуру: то есть у каждого элемента списка кроме первого есть ровно один предыдущий элемент, у каждого элемента списка кроме последнего есть ровно один следующий элемент, список не замкнут в цикл. Каждая ячейка памяти содержит информацию об одном элементе некоторого списка, каждый элемент каждого списка записан в одной из n заданных ячеек.

Формат выходных данных

Выведите n строк, i -я из строк должна содержать величины l_i и r_i — номера ячеек памяти предыдущего и следующего элемента списка для ячейки памяти i после объединения всех списков из входных данных в один. Если решений несколько, выведите любое из них.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7 | 4 7 |
| 4 7 | 5 6 |
| 5 0 | 0 5 |
| 0 0 | 6 1 |
| 6 1 | 3 2 |
| 0 2 | 2 4 |
| 0 4 | 1 0 |
| 1 0 | |

Задача В. Подготовка к сортировке слиянием

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Ивана есть массив, состоящий из n различных целых чисел. Он решил упорядочить элементы массива по возрастанию. Так как Иван является фанатом сортировки слиянием, он решил представить массив в виде одной или нескольких возрастающих последовательностей, которые затем он планирует слить в один отсортированный массив.

Представление массива в виде возрастающих последовательностей Иван формирует с помощью следующего алгоритма.

До тех пор, пока в массиве есть хотя бы одно невыписанное число, он повторяет следующую процедуру:

- просматривает массив слева направо;
- во время очередного просмотра обращает внимание только на не выписанные ранее числа;
- если рассматриваемое им в данный момент число — первое невыписанное число во время текущего просмотра, или же оно больше предыдущего выписанного во время текущего просмотра числа, то он выписывает рассматриваемое число.

Например, если массив Ивана имеет вид $[1, 3, 2, 5, 4]$, то всего он выполнит два просмотра. На первом из них он выпишет числа $[1, 3, 5]$, а на втором — $[2, 4]$.

Напишите программу, которая автоматизирует подготовительную работу Ивана — найдёт представление заданного массива в виде одной или нескольких возрастающих последовательностей в соответствии с описанным выше алгоритмом.

Формат входных данных

В первой строке содержится целое положительное число n ($1 \leq n \leq 2 \cdot 10^5$) — количество чисел в массиве Ивана.

Во второй строке содержится последовательность различных целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — массив Ивана.

Формат выходных данных

Выведите представление заданного массива в виде одной или нескольких возрастающих последовательностей в соответствии с описанным выше алгоритмом. Каждую последовательность выводите на новой строке.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------|-------------------|
| 5 1 3 2 5 4 | 1 3 5 2 4 |
| 4 4 3 2 1 | 4 3 2 1 |
| 4 10 30 50 101 | 10 30 50 101 |

Задача С. Суммарная вложенность

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Напомним, что скобочная последовательность называется правильной, если в неё возможно вставить символы '+' и '1' так, что получится правильное арифметическое выражение. Например, последовательность «`()()`» является правильной, так как из неё возможно получить правильное арифметическое выражение, вставив символы '+' и '1': «`((1+1)+(1+1))`». Аналогично, следующие последовательности — правильные: «`()()`», «`()()`» и «`()`». Следующие последовательности не являются правильными скобочными последовательностями: «`)`», «`()`» и «`()()`».

В этой задаче заданы два целых числа n и k . Постройте такую правильную скобочную последовательность из круглых скобок длины $2 \cdot n$, суммарная вложенность всех открывающих скобок в которой в точности равна k . Вложенность одной открывающей скобки равна количеству пар соответствующих скобок, в которые она вложена.

Например, в последовательности «`()()`» вложенность первой открывающей скобки равна 0, вложенность второй открывающей скобки равна 0, а вложенность третьей открывающей скобки равна 1. Таким образом, суммарная вложенность всех открывающих скобок равна 1.

Формат входных данных

В первой строке содержатся целые числа n и k ($1 \leq n \leq 3 \cdot 10^5$, $0 \leq k \leq 10^{18}$) — количество открывающих скобок и необходимая суммарная вложенность.

Формат выходных данных

Выведите построенную правильную скобочную последовательность из круглых скобок.

Если такую последовательность невозможно построить, выведите «Impossible» (без кавычек).

Примеры

| стандартный ввод | стандартный вывод |
|------------------|------------------------|
| 3 1 | <code>()()</code> |
| 4 6 | <code>((((()))</code> |
| 2 5 | Impossible |

Замечание

Первый пример разобран в условии.

Во втором примере ответ «`((((()))`». Действительно, вложенность первой открывающей скобки равна 0, вложенность второй открывающей скобки равна 1, вложенность третьей открывающей скобки равна 2, вложенность четвёртой открывающей скобки равна 3. Суммарная вложенность равна $0 + 1 + 2 + 3 = 6$.

В третьем примере невозможно построить подходящую правильную скобочную последовательность, так как максимальная возможная суммарная вложенность открывающих скобок для последовательности из двух пар скобок равна 1. Такая суммарная вложенность получается для последовательности «`()()`».

Задача D. Дог-шоу

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Очень скоро на ТВ выйдет новое шоу с участием собак и их хозяев — Дог-Шоу. На шоу от собак требуется продемонстрировать бездонный желудок, стратегическое мышление и инстинкт самосохранения. Вы и ваша собака приглашены на шоу, и, конечно же, вы хотите выиграть!

Чтобы победить на шоу, собака должна съесть как можно больше мисок с едой (в чём поможет бездонный желудок). Собаки соревнуются отдельно друг от друга по следующим правилам:

В начале шоу на прямой находится собака в точке $x = 0$, а в точках $x = 1, x = 2, \dots, x = n$ находятся n мисок с едой. Миски пронумерованы от 1 до n слева направо. Как только начинается шоу, собака начинает бежать направо к ближайшей миске.

Еда в мисках в начале шоу слишком горячая и собака не может её съесть, пока она не остынет (инстинкт самосохранения мешает этому). Про i -ю из мисок известно, что еда в ней остынет через t_i секунд после начала шоу, и в этот момент собака сможет её съесть.

Собака съедает миску еды мгновенно. Собака перемещается от точки x до точки $x + 1$ за одну секунду. При этом собаке запрещено двигаться влево, она перемещается исключительно вправо и всегда со скоростью 1 единица расстояния в секунду. Когда собака достигает миски с едой, возможны следующие варианты:

- еда уже остыла, тогда собака мгновенно съедает еду и без какой-либо остановки продолжает двигаться вправо;
- еда еще не остыла, тогда собака либо останавливается и ожидает пока еда остынет, затем съедает ее и тут же начинает двигаться вправо, либо без какой-либо остановки продолжает двигаться вправо, пропустив миску с едой.

Шоу останавливается через T секунд после начала. Если собака достигает миски с едой в момент ровно T , то она уже не успевает её съесть. Если до истечения T секунд собака убежала за самую правую из мисок шоу останавливают досрочно.

Вы должны помочь вашей собаке составить стратегию, которая позволит ей съесть максимальное количество мисок с едой за T секунд.

Формат входных данных

В первой строке даны два целых числа n и T ($1 \leq n \leq 200\,000$, $1 \leq T \leq 2 \cdot 10^9$) — количество мисок с едой и время, по истечении которого собака будет остановлена.

Во второй строке дана последовательность t_1, t_2, \dots, t_n ($1 \leq t_i \leq 10^9$), где t_i равно моменту времени, когда еда в i -й миске остынет, и собака сможет её съесть.

Формат выходных данных

Выведите одно целое число — максимальное количество мисок с едой, которые собака сможет съесть за T секунд.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 5 1 5 3 | 2 |
| 1 2 1 | 1 |
| 1 1 1 | 0 |

Замечание

В первом примере собака должна пропустить вторую миску, чтобы успеть поесть из двух — из первой и из третьей.

Задача E. Пакманы

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

Игровое поле представляет собой клетчатую полосу $1 \times n$. В некоторых клетках находятся пакманы, в некоторых клетках — звёздочки, остальные клетки пусты.

Пакман может перемещаться в соседнюю клетку за 1 единицу времени. Если в клетке находится звёздочка, то пакман в момент перемещения в эту клетку съедает звёздочку. Пакман съедает звёздочку мгновенно.

В начальный момент времени все пакманы начинают двигаться. Каждый из пакманов может неограниченное количество раз менять направление движения, но не должен выходить за пределы игрового поля. Пакманы не создают помех движению других пакманов; в одной клетке может находиться произвольное количество пакманов, движущихся в произвольных направлениях.

Ваша задача — определить минимально возможное время, за которое пакманы смогут съесть все звёздочки.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 10^5$) — длина игрового поля.

Во второй строке содержится описание игрового поля, состоящее из n символов. Если на позиции i записан символ '.', то клетка i — пуста. Если на позиции i записан символ '*', то в клетке i находится звёздочка. Если же на позиции i записан символ 'P', в клетке i находится пакман.

Гарантируется, что на игровом поле есть хотя бы один пакман и хотя бы одна звёздочка.

Формат выходных данных

Выведите минимальное время, за которое пакманы смогут съесть все звёздочки.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7 *..P*P* | 3 |
| 10 .**PP.*P.* | 2 |

Замечание

В первом примере пакман, находящийся на позиции 4, отправится влево и съест звёздочку на позиции 1. На это у него уйдёт 3 единицы времени. За эти же 3 единицы времени пакман, находящийся на позиции 6, успеет съесть обе соседние с ним звёздочки. Например, сначала он может пойти влево и съесть звёздочку на позиции 5, затратив 1 единицу времени, а потом из позиции 5 пойти вправо и съесть звёздочку на позиции 7, затратив на это ещё 2 единицы времени. Таким образом, за 3 единицы времени пакманы съедят все звёздочки на игровом поле.

Во втором примере пакман, находящийся на позиции 4, пойдёт влево и спустя 2 единицы времени уже съест звёздочки на позициях 3 и 2. Пакманы же, находящиеся на позициях 5 и 8, будут двигаться вправо и в течение 2 единиц времени съедят звёздочки на позициях 7 и 10, соответственно. Таким образом, пакманам достаточно 2 единиц времени для поедания всех звёздочек на игровом поле.

Задача F. Выборы в Берляндии

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

Сегодня — выборы в парламент Берляндии. Голосование в самом разгаре!

Всего в выборах принимают участие n кандидатов; будем считать, что они занумерованы числами от 1 до n . По результатам выборов k ($1 \leq k \leq n$) кандидатов, показавших наилучший результат на выборах, будут избраны в парламент.

Результаты выборов определяются следующим образом. По окончании голосования для каждого кандидата подсчитывается количество отданных за него голосов. Чем больше голосов набрал кандидат, тем выше место он займет в таблице результатов. Если два или более кандидатов набрали одинаковое количество голосов, то выше место в таблице результатов занимает кандидат, последний голос за которого был отдан раньше.

Таким образом, при формировании таблицы результатов кандидаты упорядочиваются по невозрастанию количества отданных за них голосов, а при равном количестве отданных за них голосов — по возрастанию времени, в которое был отдан последний голос за этого кандидата.

Если за кандидата не отдан ни один голос, то он не считается избранным ни при каких условиях. Таким образом, возможно что менее чем k кандидатов будут избраны в парламент.

В Берляндии m жителей имеют право голоса, и каждый из них обязательно примет участие в голосовании. Каждый из жителей отдаст свой голос за одного из n кандидатов. Варианта «против всех» в бюллетене нет, портить бюллетени или не ходить на выборы в Берляндии не принято. Таким образом, каждый из m жителей должен проголосовать за одного из n кандидатов.

К настоящему моменту уже проголосовали a ($1 \leq a \leq m$) жителей Берляндии. Бюллетени в Берляндии именные и мгновенно обрабатываются в момент голосования, поэтому для j -го проголосовавшего уже известен номер кандидата g_j , за которого он отдал свой голос. Проголосовавшие жители нумеруются в хронологическом порядке; то есть $(j + 1)$ -й житель голосовал позже j -го.

Оставшиеся $m - a$ жителей проголосуют до окончания выборов, каждый из них отдаст голос за какого-то из n кандидатов.

Ваша задача — определить для каждого из n кандидатов один из трёх возможных исходов:

- кандидат уже может считаться избранным в парламент независимо от распределения голосов оставшихся $m - a$ жителей;
- кандидат имеет возможность быть избранным в парламент после подведения итогов выборов;
- кандидат не может быть избранным в парламент независимо от распределения голосов оставшихся $m - a$ жителей.

Формат входных данных

В первой строке содержатся четыре целых числа n , k , m и a ($1 \leq k \leq n \leq 100$, $1 \leq m \leq 100$, $1 \leq a \leq m$) — количество кандидатов, количество мест в парламенте, количество жителей Берляндии и количество уже проголосовавших жителей.

Во второй строке содержится последовательность из a целых чисел g_1, g_2, \dots, g_a ($1 \leq g_j \leq n$), где g_j — номер кандидата, за которого отдал голос j -й из уже проголосовавших жителей. Проголосовавшие жители нумеруются в порядке возрастания времени голосования.

Формат выходных данных

Выведите последовательность из n целых чисел r_1, r_2, \dots, r_n , где:

- $r_i = 1$, если i -й кандидат гарантированно будет избран в парламент независимо от распределения голосов оставшихся $m - a$ жителей;
- $r_i = 2$, если i -й кандидат имеет шанс быть избранным в парламент, то есть оставшиеся $m - a$ жителей могут проголосовать таким образом, что он будет избран в парламент;

- $r_i = 3$, если i -й кандидат не будет избран в парламент независимо от распределения голосов оставшихся $m - a$ жителей.

Примеры

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

Задача G. Пары в университете

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

В университете учится n групп студентов. В течение учебного дня у каждой группы может состояться не более 7 пар. Для этого выделено семь временных слотов, пронумерованных от 1 до 7.

Для каждой группы известно её расписание на понедельник, то есть известны временные слоты, в которые будут проходить пары у этой группы.

Ваша задача — определить минимальное количество аудиторий, необходимых для проведения пар у всех групп в понедельник. Учитывайте, что в один и тот же слот в одной аудитории может заниматься не более одной группы.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 1000$) — количество групп.

В каждой из следующих n строк содержится последовательность из 7 нулей и единиц — расписание занятий в понедельник для очередной группы. Если символ на некоторой позиции равен 1, то у группы есть пара в соответствующий временной слот. В противном случае, этой пары у группы нет.

Формат выходных данных

Выведите минимальное количество аудиторий, необходимых для проведения пар у всех групп в понедельник.

Примеры

| стандартный ввод | стандартный вывод |
|------------------------------------|-------------------|
| 2 0101010 1010101 | 1 |
| 3 0101011 0011001 0110111 | 3 |

Замечание

В первом примере достаточно одной аудитории, которая будет занята в каждый из семи временных слотов либо первой группой, либо второй.

Во втором примере нужно три аудитории, так как в седьмой временной слот пара есть у всех трёх групп.

Задача N. Нагрузочное тестирование

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

Поликарп планирует провести нагрузочное тестирование своего нового проекта Fakebook. Он уже договорился с друзьями, чтобы в определённые моменты времени они посылали запросы в Fakebook. Тестирование продлится n минут, в i -ю минуту друзья пошлют a_i запросов.

Поликарп планирует протестировать Fakebook под особым видом нагрузки. В случае попадания информации о Fakebook в СМИ Поликарп надеется на монотонный рост нагрузки, который сменится монотонным убыванием интереса к сервису. Такой профиль нагрузки и собирается протестировать Поликарп во время нагрузочного теста.

Какое минимальное количество запросов придется дополнительно сделать Поликарпу, чтобы сначала нагрузка на сервер строго возрастала, а затем строго убывала? Как возрастающая часть, так и убывающая часть могут быть пустыми (то есть отсутствовать). Убывание должно сразу же сменить возрастание, например, участок одинаковых подряд значений нагрузки недопустим.

К примеру, если нагрузка описывается одним из массивов $[1, 2, 8, 4, 3]$, $[1, 3, 5]$ или $[10]$, то такой профиль нагрузки устраивает Поликарпа (в каждом из этих случаев сначала идёт участок строгого возрастания, затем тут же строго убывания). Если нагрузка описывается одним из массивов $[1, 2, 2, 1]$, $[2, 1, 2]$ или $[10, 10]$, то такой профиль нагрузки не устраивает Поликарпа.

Помогите Поликарпу сделать минимальное количество дополнительных запросов, чтобы получившийся профиль нагрузки его устраивал. Дополнительные запросы Поликарп может делать в любом количестве в любые минуты от 1 до n .

Формат входных данных

В первой строке содержится целое положительное число n ($1 \leq n \leq 100\,000$) — продолжительность нагрузочного тестирования.

Во второй строке содержатся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), где a_i — количество запросов со стороны друзей в i -ю минуту тестирования.

Формат выходных данных

Выведите минимальное количество дополнительных запросов, которые надо сделать Поликарпу, чтобы нагрузка сначала строго возрастала, а затем сразу строго убывала.

Примеры

| стандартный ввод | стандартный вывод |
|---------------------------|-------------------|
| 5 1 4 3 2 5 | 6 |
| 5 1 2 2 2 1 | 1 |
| 7 10 20 40 50 70 90 30 | 0 |

Замечание

В первом примере Поликарп должен сделать два дополнительных запроса в третью минуту и четыре — в четвёртую. Тогда получившаяся нагрузка примет вид: $[1, 4, 5, 6, 5]$. Суммарно будет сделано 6 дополнительных запросов.

Во втором примере достаточно один раз сделать дополнительный запрос в третью минуту, поэтому ответ равен 1.

В третьем примере нагрузка уже удовлетворяет всем описанным в условии требованиям, поэтому ответ 0.

Задача I. Уровень шума

Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Столица Берляндии имеет вид прямоугольника размером $n \times m$ кварталов. Все кварталы делятся на три типа:

- обычные (обозначаются символом \cdot) — такие кварталы сами не производят шума, но не являются препятствиями для распространения шума;
- источники шума (обозначаются заглавными латинскими буквами от 'A' до 'Z') — такие кварталы являются источниками шума и не являются препятствиями для распространения шума;
- сильно застроенные (обозначаются символом $*$) — такие кварталы являются шумоизолированными, в них не проникает шум, и они сами являются препятствиями для распространения шума.

Квартал, который обозначается буквой 'A', производит q единиц шума. Квартал, который обозначается буквой 'B', производит $2 \cdot q$ единиц шума. И так далее, вплоть до квартала, обозначаемого буквой 'Z', который производит $26 \cdot q$ единиц шума. В городе может располагаться произвольное количество кварталов, обозначаемых одинаковыми буквами.

При распространении от источника при переходе к соседнему по стороне кварталу шум затихает в два раза (при делении следует округлять вниз). Шум распространяется по цепочке. Например, если квартал расположен на расстоянии 2 от источника шума, то величина шума, который дойдёт до квартала, делится на 4. Таким образом, уровень шума, дошедший от источника к кварталу определяется исключительно длиной кратчайшего пути между ними. Сильно застроенные кварталы являются препятствиями, шум в них не проникает.

| | | | | | | |
|---------|---------|---------|---------------|--------------|--------------|--------------|
| \cdot | \cdot | \cdot | | $50+25=75$ | $25+12=37$ | $12+25=37$ |
| A | * | \cdot | \Rightarrow | $100+50=150$ | 0 | $6+50=56$ |
| \cdot | B | \cdot | | $50+100=150$ | $25+200=225$ | $12+100=112$ |

Значения в ячейках таблицы справа показывают суммарный уровень шума в соответствующих кварталах для $q = 100$, первое слагаемое — шум от квартала 'A', второе слагаемое — шум от квартала 'B'.

Уровень шума в квартале определяется как сумма шумов от всех источников. Для оценки качества жизни населения столицы Берляндии требуется найти количество кварталов, уровень шума в которых превышает допустимую норму p .

Формат входных данных

В первой строке содержатся четыре целых числа n , m , q и p ($1 \leq n, m \leq 250$, $1 \leq q, p \leq 10^6$) — размеры столицы Берляндии, количество единиц шума, которое производит квартал 'A', и допустимая норма шума.

В следующих n строках содержится по m символов — описание кварталов столицы в соответствии с условием. Возможно, что в столице Берляндии отсутствуют кварталы любого из трёх типов.

Формат выходных данных

Выведите количество кварталов, уровень шума в которых превышает допустимую норму p .

Примеры

| стандартный ввод | стандартный вывод |
|----------------------------------|-------------------|
| 3 3 100 140 ... A*. .B. | 3 |
| 3 3 2 8 B*. BB* BBB | 4 |
| 3 4 5 4 ..*B ..** D... | 7 |

Замечание

Иллюстрация к первому примеру находится в основной части условия задачи.

Задача J. Посвящение в студенты

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Скоро в Берляндском университете состоится посвящение первокурсников в студенты. Организаторы посвящения придумывают программу этого праздника. По их мнению, было бы хорошо, если бы новоиспечённые студенты подарили друг другу небольшие сувениры. Когда они озвучили эту идею первокурсникам, то выяснили следующее:

- некоторые пары первокурсников уже знакомы друг с другом;
- каждый первокурсник согласен дарить сувениры только тем, с кем он уже знаком;
- каждый первокурсник не хотел бы дарить много сувениров.

Организаторы записали все пары знакомых между собой первокурсников и теперь хотят определить для каждого первокурсника, кому он должен подарить сувениры. По мнению организаторов, в каждой паре знакомых между собой первокурсников *ровно один* должен подарить сувенир другому.

Первокурсники уже решили назвать самым невезучим того, кому придётся дарить наибольшее количество сувениров. Организаторы в ответ пообещали, что самый невезучий будет минимально невезучим: конечно, ему придётся дарить наибольшее количество сувениров по сравнению с остальными, но это количество будет минимально возможным.

Организаторам очень некогда, и они обратились к вам с просьбой определить для каждой пары знакомых первокурсников, кто кому должен подарить сувенир.

Формат входных данных

В первой строке содержатся два целых числа n и m ($1 \leq n \leq 5000$, $0 \leq m \leq \min(5000, n \cdot (n-1)/2)$) — количество первокурсников и количество пар первокурсников, знакомых между собой. Студенты пронумерованы от 1 до n .

В каждой из следующих m строк содержится по два целых числа x_i, y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$) — номера первокурсников, знакомых между собой.

Гарантируется, что любая пара содержится в списке единожды. Также гарантируется, что если в списке содержится пара (x_i, y_i) , в нём не содержится пара (y_i, x_i) .

Формат выходных данных

В первой строке выведите единственное целое число — минимальное количество сувениров, которые подарит самый невезучий первокурсник.

В каждой из следующих m строк выведите по одной паре номеров знакомых между собой первокурсников. Первым в паре выводите номер того первокурсника, который будет дарить сувенир.

Пары можно выводить в любом порядке. Если существует несколько решений, выведите любое из них.

Примеры

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

Задача К. Проездные билеты

Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вечером Поликарп решил проанализировать свои сегодняшние расходы на поездки в общественном транспорте.

Автобусная сеть в столице Берляндии устроена таким образом, что каждый автобус курсирует по маршруту между двумя конечными остановками. Промежуточных остановок у автобуса нет. Таким образом, каждый из автобусов непрерывно курсирует по маршруту от одной конечной остановки до другой и назад. Между парой остановок курсирует не более одного автобуса.

Поликарп совершил n поездок на автобусе, каждая из которых задается названием остановки, откуда он уехал, и названием остановки, куда он приехал. В записях Поликарпа поездки следуют в хронологическом порядке.

Известно, что одна поездка на автобусе любого маршрута стоит a бурлей. Однако, если пассажир совершает пересадку, то цена поездки уменьшается до b бурлей ($b < a$). Считается, что пассажир совершает пересадку, если остановка, на которой он садится в автобус, совпадает с остановкой, на которой он вышел из предыдущего автобуса. Очевидно, что первая поездка не может быть совершена после пересадки.

Например, если Поликарп совершил последовательно три поездки: «BerBank» → «University», «University» → «BerMall», «University» → «BerBank», то он заплатит $a + b + a = 2a + b$ бурлей. Из BerBank он приехал в University, там пересел и поехал в BerMall. Затем он прошёл пешком до University и вернулся в BerBank.

Поликарп мог купить не более чем k проездных по цене f бурлей за один проездной. Проездной покупается для одного автобусного маршрута и делает любую поездку по этому маршруту (в любом направлении) бесплатной. Единожды купленный проездной может использоваться неограниченное количество раз любым из двух направлений.

Какую наименьшую сумму денег мог потратить Поликарп сегодня, если бы приобрел не более k проездных оптимальным образом?

Формат входных данных

В первой строке записаны пять целых чисел n, a, b, k, f ($1 \leq n \leq 300, 1 \leq b < a \leq 100, 0 \leq k \leq 300, 1 \leq f \leq 1000$), где:

- n — количество поездок Поликарпа,
- a — цена одной обычной поездки,
- b — цена одной поездки после пересадки,
- k — максимальное количество проездных,
- f — цена одного проездного.

Далее следуют n строк, описывающие поездки в порядке их совершения. Каждая строка содержит ровно два различных слова, разделенных единичным пробелом — название стартовой остановки и название финальной остановки для очередной поездки Поликарпа. Все названия состоят из прописных и строчных букв латинского алфавита, содержат от 1 до 20 букв. Прописные и строчные буквы следует считать различными.

Формат выходных данных

Выведите наименьшую сумму денег, которую Поликарп мог потратить сегодня, если бы приобрел не более k проездных оптимальным образом.

Примеры

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 3 5 3 1 8 BerBank University University BerMall University BerBank | 11 |
| 4 2 1 300 1000 a A A aa aa AA AA a | 5 |

Замечание

В первом примере Поликарп может купить проездной на маршрут «BerBank \longleftrightarrow University», потратив 8 бурлей. Заметим, что его вторая поездка «University» \rightarrow «BerMall» совершается после пересадки; следовательно, за эту поездку он заплатит 3 бурля. Таким образом, искомая сумма денег равна $8 + 3 = 11$ бурлей.

Во втором примере Поликарпу невыгодно покупать проездные. Заметим, что каждая его поездка, кроме первой, совершена после пересадки. Следовательно, искомая сумма денег равна $2+1+1+1 = 5$ бурлей.

Задача L. Компьютерная сеть Берляндского ГУ

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В компьютерной сети Берляндского государственного университета имеется n маршрутизаторов, пронумерованных от 1 до n . Некоторые пары маршрутизаторов соединены коммутационными шнурами (патч-кордами). Информация по патч-корду может передаваться в любом из двух направлений. Сеть устроена таким образом, что между любыми двумя маршрутизаторами возможна передача информации (непосредственно или через другие маршрутизаторы). В сети отсутствуют циклы, поэтому между каждой парой маршрутизаторов существует единственный путь по патч-кордам.

К сожалению, точная топология сети была утеряна администраторами. С целью её восстановления была собрана следующая вспомогательная информация.

Для каждого патч-корда p , непосредственно присоединённого к маршрутизатору i , известны все маршрутизаторы, находящиеся за патч-кордом p относительно i . Иными словами, известны все маршрутизаторы, путь от которых до маршрутизатора i включает в себя патч-корд p . Таким образом, для каждого маршрутизатора i составлены k_i списков, где k_i — количество патч-кордов, подключенных к i .

Например, пусть сеть состоит из трех маршрутизаторов, соединённых в цепочку 1 – 2 – 3. Тогда:

- маршрутизатор 1: для единственного патч-корда, присоединённого к первому маршрутизатору, составлен единственный список, который содержит два маршрутизатора: 2 и 3;
- маршрутизатор 2: для каждого из патч-кордов, присоединённых ко второму маршрутизатору, составлено по списку: один из них содержит маршрутизатор 1, а другой — маршрутизатор 3;
- маршрутизатор 3: для единственного патч-корда, присоединённого к третьему маршрутизатору, составлен один список, который содержит два маршрутизатора: 1 и 2.

Помогите администраторам компьютерной сети БГУ восстановить топологию сети, то есть определить все пары маршрутизаторов, непосредственно соединенные патч-кордами.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 1000$) — количество маршрутизаторов в сети БГУ.

В каждой из следующих n строк содержатся описания списков для очередного маршрутизатора. Так, i -я строка содержит списки для i -го маршрутизатора.

Описание каждого списка начинается с количества маршрутизаторов в нём. Затем следует символ ‘:’, а после него через запятую перечисляются номера всех маршрутизаторов из списка. Описания списков для каждого патч-корда разделяются символом ‘-’.

Гарантируется, что для каждого маршрутизатора суммарное количество маршрутизаторов во всех списках равно $n - 1$, причём все числа в списках каждого маршрутизатора различны. Списки каждого маршрутизатора не содержат номер этого маршрутизатора.

Формат выходных данных

Если решения не существует, выведите -1.

В противном случае в первую строку выведите число $n - 1$ — количество патч-кордов в сети. В каждой из следующих $n - 1$ строк выведите по два целых числа — описания патч-кордов в виде номеров соединяемых ими маршрутизаторов. Информация о каждом патч-корде должна быть выведена ровно один раз.

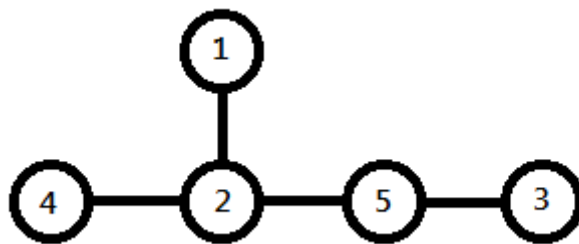
Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------------------|
| 3 2:3,2 1:1-1:3 2:1,2 | 2 2 1 2 3 |
| 5 4:2,5,3,4 1:4-1:1-2:5,3 4:4,5,2,1 4:2,1,3,5 1:3-3:4,2,1 | 4 2 1 2 4 5 2 3 5 |
| 3 1:2-1:3 1:1-1:3 1:1-1:2 | -1 |

Замечание

Первый пример разобран в условии.

Ответ для второго примера изображён на рисунке.



На нём мы можем видеть, что у первого маршрутизатора один список, в котором встречаются все остальные маршрутизаторы. У второго маршрутизатора три списка: в первом — маршрутизатор 4, во втором — маршрутизатор 1, а в третьем — маршрутизаторы 3 и 5. У третьего маршрутизатора один список, в котором встречаются все остальные маршрутизаторы. У четвёртого маршрутизатора тоже один список, в котором встречаются все остальные маршрутизаторы. У пятого маршрутизатора два списка: в первом — маршрутизатор 3, а во втором — маршрутизаторы 1, 2 и 4.

Задача М. Погода на завтра

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

Вася придумал свою систему прогнозирования погоды. Он знает информацию о средней температуре воздуха в каждый из последних n дней. Считайте, что средняя температура в каждый из дней — целое число.

Вася считает, что если средние температуры за последние n дней составляют арифметическую прогрессию, в которой первый член равен средней температуре в первый день, второй член равен средней температуре во второй день и так далее, то температура в следующий, $(n + 1)$ -й день будет равна следующему члену арифметической прогрессии. В противном случае, как считает Вася, температура в $(n + 1)$ -й день будет равна температуре в n -й день.

Перед вами стоит задача помочь Васе предсказать среднюю температуру на завтра, то есть в $(n + 1)$ -й день.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 100$) — количество дней, для которых известна информация о средней температуре.

Во второй строке содержится последовательность целых чисел t_1, t_2, \dots, t_n ($-1000 \leq t_i \leq 1000$), где t_i — средняя температура воздуха в день i .

Формат выходных данных

Выведите среднюю температуру воздуха в день $(n + 1)$, которую предскажет Вася согласно своему алгоритму. Обратите внимание, что абсолютная величина предсказанной температуры может превышать 1000.

Примеры

| стандартный ввод | стандартный вывод |
|--------------------|-------------------|
| 5 10 5 0 -5 -10 | -15 |
| 4 1 1 1 1 | 1 |
| 3 5 1 -5 | -5 |
| 2 900 1000 | 1100 |

Замечание

В первом примере последовательность средних температур представляет собой арифметическую прогрессию, в которой первый член равен 10, а каждый следующий уменьшается на 5. Поэтому предсказанная температура в шестой день $-10 - 5 = -15$.

Во втором примере последовательность средних температур представляет собой арифметическую прогрессию, в которой первый член равен 1, а каждый следующий равен предыдущему. Поэтому предсказанная температура в пятый день также будет равна 1.

В третьем примере средние температуры не составляют арифметическую прогрессию, поэтому температура в четвёртый день равна температуре в третий день, то есть -5 .

В четвёртом примере последовательность средних температур представляет собой арифметическую прогрессию, в которой первый член равен 900, а следующий увеличен на 100. Поэтому предсказанная температура в третий день $1000 + 100 = 1100$.