Оглавление

[Условия задач по алгоритмам 2](#_Toc174273121)

[Яндекс контест по предмету “Алгоритмы и Структуры Данных” 2](#_Toc174273122)

[Задача №1 (Агроном-Любитель) 2](#_Toc174273123)

[Задача №2 (Зоопарк Глеба) 2](#_Toc174273124)

[Задача №3 (Конфигурационный файл) 3](#_Toc174273125)

[Задача №4 (профессор Хаос) 4](#_Toc174273126)

[Задача №5 (Коровы в стойла) 4](#_Toc174273127)

[Задача №6 (Число) 5](#_Toc174273128)

[Задача №7 (Кошмар в замке) 5](#_Toc174273129)

[Задача №8 (Магазин) 6](#_Toc174273130)

[Задача №9 (Машинки) 6](#_Toc174273131)

[Задача №10 (Гоблины и очереди) 7](#_Toc174273132)

[Задача №11 (Менеджер памяти-1) 7](#_Toc174273133)

[Задача №12 (Минимум на отрезке) 8](#_Toc174273134)

[Задача №13 (Цивилизация) 8](#_Toc174273135)

[Задача №14 (Свинки-Копилки) 9](#_Toc174273136)

[Задача №15 (Долой списывание!) 9](#_Toc174273137)

[Задача №16 (Авиаперелёты) 10](#_Toc174273138)

[Тимус 10](#_Toc174273139)

[1005 (Куча камней) 10](#_Toc174273140)

[1444 (Накормить Элефпотама) 11](#_Toc174273141)

[1450 (Российские газопроводы) 11](#_Toc174273142)

[1628 (Белые полосы) 12](#_Toc174273143)

[1650 (Миллиардеры) 13](#_Toc174273144)

[1726 (Кто ходит в гости…) 13](#_Toc174273145)

# **Условия задач по алгоритмам**

## **Яндекс контест по предмету “Алгоритмы и Структуры Данных”**

### ***Задача №1 (Агроном-Любитель)***

Городской школьник Лёша поехал на лето в деревню и занялся выращиванием цветов. Он посадил *n* цветков вдоль одной длинной прямой грядки, и они успешно выросли. Лёша посадил множество различных видов цветков, *i*-й от начала грядки цветок имеет вид *ai*, где *ai* "— целое число, номер соответствующего вида в «Каталоге юного агронома».

Теперь Лёша хочет сделать фотографию выращенных им цветов и выложить ее в раздел «мои грядки» в социальной сети для агрономов «ВКомпосте». На фотографии будет виден отрезок из одного или нескольких высаженных подряд цветков.

Однако он заметил, что фотография смотрится не очень интересно, если на ней много одинаковых цветков подряд. Лёша решил, что если на фотографии будут видны три цветка одного вида, высаженные подряд, то его друзья — специалисты по эстетике цветочных фотографий — поставят мало лайков.

Помогите ему выбрать для фотографирования как можно более длинный участок своей грядки, на котором нет трех цветков одного вида подряд.

**Формат ввода**

В первой строке содержится целое число *n* (*1 ≤ n ≤ 200 000*) — количество цветов на грядке.

Во второй строке содержится *n* целых чисел *ai* (*1 ≤ ai ≤ 109*), обозначающих вид очередного цветка. Одинаковые цветки обозначаются одинаковыми числами, разные — разными.

**Формат вывода**

Выведите номер первого и последнего цветка на самом длинном искомом участке. Цветки нумерeются от 1 до *n*.

Если самых длинных участков несколько, выведите участок, который начинается раньше.

### ***Задача №2 (Зоопарк Глеба)***

Недавно Глеб открыл зоопарк. Он решил построить его в форме круга и, естественно, обнёс забором. Глеб взял вас туда начальником охраны. Казалось бы все началось так хорошо, но именно в вашу первую смену все животные разбежались. В зоопарке *n* животных различных видов, также под каждый из видов есть свои ловушки. К сожалению некоторые животные враждуют с друг другом в природе (они обозначены разными буквами), а зоопарк обнесён забором и имеет форму круга. С помощью камер, удалось выяснить, где находятся все животные. Умная система поддержки жизнедеятельности зоопарка уже просканировала зоопарк и вывела id всех животных и ловушек в том порядке, в котором они видны из центра зоопарка. Получилось так, что все животные и все ловушки находятся на краю зоопарка. Вы хотите понять, могут ли животные прийти в свою ловушку так, чтобы их путь не пересекался с другими. Если да, также предъявите какую-нибудь из схем поимки животных.

**Формат ввода**

 На вход подается строчка из *2 ⋅ n* символов латинского алфавита, где маленькая буква - животное, а большая - ловушка. Размер строки не более *100000*.

**Формат вывода**

Требуется вывести "Impossible", если решения не существует или "Possible", если можно вернуть всех животных в клетки. В случае если можно, то для каждой ловушки в порядке обхода требуется вывести индекс животного в ней.

### ***Задача №3 (Конфигурационный файл)***

Вадим разрабатывает парсер конфигурационных файлов для своего проекта. Файл состоит из блоков, которые выделяются с помощью символов «{» — начало блока, и «}» — конец блока. Блоки могут вкладываться друг в друга. В один блок может быть вложено несколько других блоков.

В конфигурационном файле встречаются переменные. Каждая переменная имеет имя, которое состоит из не более чем десяти строчных букв латинского алфавита. Переменным можно присваивать числовые значения. Изначально все переменные имеют значение *0*.

Присваивание нового значения записывается как <variable>=<number>, где <variable> — имя переменной, а <number> — целое число, по модулю не превосходящее *109*. Парсер читает конфигурационный файл построчно. Как только он встречает выражение присваивания, он присваивает новое значение переменной. Это значение сохраняется до конца текущего блока, а затем восстанавливается старое значение переменной. Если в блок вложены другие блоки, то внутри тех из них, которые идут после присваивания, значение переменной также будет новым.

Кроме того, в конфигурационном файле можно присваивать переменной значение другой переменной. Это действие записывается как <variable1>=<variable2>. Прочитав такую строку, парсер присваивает текущее значение переменной variable2 переменной variable1. Как и в случае присваивания константного значения, новое значение сохраняется только до конца текущего блока. После окончания блока переменной возвращается значение, которое было перед началом блока.

Для отладки Вадим хочет напечатать присваиваемое значение для каждой строки вида <variable1>=<variable2>. Помогите ему отладить парсер.

**Формат ввода**

Входные данные содержат хотя бы одну и не более *105* строк. Каждая строка имеет один из четырех типов:

* { — начало блока;
* } — конец блока;
* <variable>=<number> — присваивание переменной значения, заданного числом;
* <variable1>=<variable2> — присваивание одной переменной значения другой переменной. Переменные <variable1> и <variable2> могут совпадать.

Гарантируется, что ввод является корректным и соответствует описанию из условия. Ввод не содержит пробелов.

**Формат вывода**

Для каждой строки типа <variable1>=<variable2> выведите значение, которое было присвоено.

### ***Задача №4 (профессор Хаос)***

В секретной лаборатории профессора Хаоса проходит эксперимент по выращиванию особо опасных бактерий. В начале первого дня эксперимента у Хаоса имеется *a* особо опасных бактерий.

Каждый день эксперимента устроен следующим образом. Рано утром профессор достает из контейнера все свои бактерии и помещает их в инкубатор, где бактерии начинают делиться. Вместо каждой бактерии образуется *b* новых бактерий.

После извлечения бактерий из инкубатора *c* из них используются для проведения различных опытов и затем уничтожаются. Если после извлечения из инкубатора имеется менее *c* бактерий, для проведения опытов используются все имеющиеся бактерии, и эксперимент заканчивается.

Оставшиеся бактерии в конце дня необходимо поместить в контейнер и продолжить использовать в эксперименте. Однако в контейнер можно поместить не более *d* бактерий, поэтому если число оставшихся бактерий больше *d*, то в контейнер помещаются *d* бактерий, а остальные уничтожаются.

Теперь профессор Хаос хочет выяснить, сколько особо опасных бактерий будет у него в контейнере после *k*-го дня эксперимента. Помогите ему найти ответ на этот вопрос.

**Формат ввода**

В единственной строке входного файла содержится пять целых чисел *a*, *b*, *c*, *d* и *k* (*1 ≤ a, b ≤ 1000*, *0 ≤ c ≤ 1000*, *1 ≤ d ≤ 1000*, *a ≤ d*, *1 ≤ k ≤ 1018*).

**Формат вывода**

Выведите одно число — количество бактерий у Хаоса к концу *k*-го дня. Если эксперимент завершится в *k*-й день или ранее, выведите число 0.

### ***Задача №5 (Коровы в стойла)***

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное расcтояние между коровами было как можно больше.

**Формат ввода**

В первой строке вводятся числа *N* *(2 < N ≤ 105)* – количество стойл и *K* *(1 < K < N )* – количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания – координаты стойл (координаты не превосходят *109*).

**Формат вывода**

Выведите одно число – наибольшее возможное допустимое расстояние.

### ***Задача №6 (Число)***

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрезала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

**Формат ввода**

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

**Формат вывода**

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

### ***Задача №7 (Кошмар в замке)***

Ходят легенды, что пока Аврора спала, ей снилось, что она ходит по разным местам: леса, поля, города и сёла. И вот однажды она наткнулась на пещеру, в которой сидел мудрец. Когда мудрец поднял на Аврору глаза, он изрёк: «*Дорогая Аврора! Ты уже годами скитаешься по этим землям. Я хочу предложить тебе задачку. Вот тебе строка s. Каждая буква из алфавита имеет свой вес ci. Вес строки, которую ты можешь получить из s многократным обменом любых двух букв, вычисляется так: для каждой буквы алфавита посчитай максимальное расстояние между позициями, в которых стоит эта буква и перемножь его с весом этой буквы. Принеси мне строку максимально возможного веса, и я тебе расскажу, в чём смысл жизни*».

К счастью, когда Аврора уже шла со строкой к мудрецу, её поцеловал Филипп, и Аврора вышла из этого кошмара. Теперь вам предлагается самим окунуться в этот кошмар и решить поставленную задачу.

**Формат ввода**

Дана строка, состоящая из строчных букв латинского алфавита *(1 ≤ |s| ≤ 105)*. Следующая строка ввода содержит 26 чисел — веса букв латинского алфавита от «a» до «z», веса неотрицательны и не превосходят *231 - 1*.

**Формат вывода**

Выведите строку *s*, в которой переставлены буквы так, чтобы полученный вес был максимально возможным. Если искомых вариантов несколько, выведите любой из них.

### ***Задача №8 (Магазин)***

У Билла большая семья: трое сыновей, девять внуков. И всех надо кормить. Поэтому Билл раз в неделю ходит в магазин.

Однажды Билл пришел в магазин и увидел, что в магазине проводится акция под названием «каждый *k*-й товар бесплатно». Изучив правила акции, Билл выяснил следующее. Пробив на кассе товары, покупатель получает чек. Пусть в чеке *n* товаров, тогда *n/k* округленное вниз самых дешевых из них достаются бесплатно.

Например, если в чеке пять товаров за 200, 100, 1000, 400 и 100 рублей, соответственно, и *k = 2*, то бесплатно достаются оба товара по 100 рублей, всего покупатель должен заплатить 1600 рублей.

Билл уже выбрал товары, и направился к кассе, когда сообразил, что товары, которые он хочет купить, можно разбить на несколько чеков, и благодаря этому потратить меньше денег.

Помогите Биллу выяснить, какую минимальную сумму он сможет заплатить за выбранные товары, возможно разбив их на несколько чеков.

**Формат ввода**

Первая строка входного файла содержит два целых числа *n*, *k* (*1 ≤ n ≤ 100 000*, *2 ≤ k ≤ 100*) — количество товаров, которые хочет купит Билл и параметр акции «каждый *k*-й товар бесплатно».

Следующая строка содержит *n* целых чисел *ai* (*1 ≤ ai ≤ 10 000*) — цены товаров, которые покупает Билл.

### ***Задача №9 (Машинки)***

Петя, которому три года, очень любит играть с машинками. Всего у Пети *N* различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более *K* машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиной игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

**Формат ввода**

В первой строке содержаться три числа *N*, *K* и *P* (*1≤ K, N ≤ 100000*, *1≤ P ≤ 500000*). В следующих *P* строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

**Формат вывода**

Выведите единственное число: минимальное количество операций, которое надо совершить Петиной маме.

### ***Задача №10 (Гоблины и очереди)***

Гоблины Мглистых гор очень любях ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

**Формат ввода**

В первой строке входных данный записано число *N* *(1 ≤ N ≤ 105)*  количество запросов. Следующие *N* строк содержат описание запросов в формате:

* + *i*  гоблин с номером *i* *(1 ≤ i ≤ N)* встаёт в конец очереди.
* \* *i*  привилегированный гоблин с номером i встает в середину очереди.
* -  первый гоблин из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

**Формат вывода**

Для каждого запроса типа - программа должна вывести номер гоблина, который должен зайти к шаманам.

### ***Задача №11 (Менеджер памяти-1)***

Пете поручили написать менеджер памяти для новой стандартной библиотеки языка \varphi++. В распоряжении у менеджера находится массив из *N* последовательных ячеек памяти, пронумерованных от *1* до *N*. Задача менеджера – обрабатывать запросы приложений на выделение и освобождение памяти. Запрос на выделение памяти имеет один параметр *K*. Такой запрос означает, что приложение просит выделить ему *K* последовательных ячеек памяти. Если в распоряжении менеджера есть хотя бы один свободный блок из *K* последовательных ячеек, то он обязан в ответ на запрос выделить такой блок. При этом непосредственно перед самой первой ячейкой памяти выделяемого блока не должно располагаться свободной ячейки памяти. После этого выделенные ячейки становятся занятыми и не могут быть использованы для выделения памяти, пока не будут освобождены. Если блока из *K* последовательных свободных ячеек нет, то запрос отклоняется. Запрос на освобождение памяти имеет один параметр *T*. Такой запрос означает, что менеджер должен освободить память, выделенную ранее при обработке запроса с порядковым номером *T*. Запросы нумеруются, начиная с единицы. Гарантируется, что запрос с номером *T* – запрос на выделение, причем к нему еще не применялось освобождение памяти. Освобожденные ячейки могут снова быть использованы для выделения памяти. Если запрос с номером *T* был отклонен, то текущий запрос на освобождение памяти игнорируется. Требуется написать менеджер памяти, удовлетворяющий приведенным критериям.

**Формат ввода**

Первая строка входного файла содержит числа *N* и *M* – количество ячеек памяти и количество запросов соответственно (*1 ≤ N ≤ 231 - 1*; *1 ≤ M ≤ 105*). Каждая из следующих *M* строк содержит по одному числу: (*i+1*)-я строка входного файла (*1 ≤ i ≤ M*) содержит либо положительное число *K*, если *i*-й запрос – запрос на выделение с параметром *K* (*1 ≤ K ≤ N*), либо отрицательное число *-T*, если *i*-й запрос – запрос на освобождение с параметром *T* (*1 ≤ T < i*).

**Формат вывода**

Для каждого запроса на выделение памяти выведите в выходной файл результат обработки этого запроса: для успешных запросов выведите номер первой ячейки памяти в выделенном блоке, для отклоненных запросов выведите число *-1*. Результаты нужно выводить в порядке следования запросов во входном файле.

### ***Задача №12 (Минимум на отрезке)***

Рассмотрим последовательность целых чисел длины *N*. По ней с шагом *1* двигается «окно» длины *K*, то есть сначала в «окне» видно первые *K* чисел, на следующем шаге в «окне» уже будут находиться *K* чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

**Формат ввода**

В первой строке входных данных содержатся два числа *N* и *K* (*1 ≤ N ≤ 150000*, *1 ≤ K ≤ 10000*, *K ≤ N*) – длины последовательности и «окна», соответственно. На следующей строке находятся *N* чисел – сама последовательность. Числа последовательности не превосходят по модулю *105*.

**Формат вывода**

Выходые данные должны содержать *N - K + 1* строк – минимумы для каждого положения «окна».

### ***Задача №13 (Цивилизация)***

Карта мира в компьютерной игре «Цивилизация» версии *1* представляет собой прямоугольник, разбитый на квадратики. Каждый квадратик может иметь один из нескольких возможных рельефов, для простоты ограничимся тремя видами рельефов  — поле, лес и вода. Поселенец перемещается по карте, при этом на перемещение в клетку, занятую полем, необходима одна единица времени, на перемещение в лес  — две единицы времени, а перемещаться в клетку с водой нельзя.

У вас есть один поселенец, вы определили место, где нужно построить город, чтобы как можно скорее завладеть всем миром. Найдите маршрут переселенца, приводящий его в место строительства города, требующий минимального времени. На каждом ходе переселенец может перемещаться в клетку, имеющую общую сторону с той клеткой, где он сейчас находится.

**Формат ввода**

Во входном файле записаны два натуральных числа *N* и *M*, не превосходящих *1000*  — размеры карты мира (*N*  — число строк в карте, *M*  — число столбцов). Затем заданы координаты начального положения поселенца *x* и *y*, где *x*  — номер строки, *y*  — номер стролбца на карте (*1 ≤ x ≤ N*, *1 ≤ y ≤ M*), строки нумеруются сверху вниз, столбцы  — слева направо. Затем аналогично задаются координаты клетки, куда необходимо привести поселенца.

Далее идет описание карты мира в виде *N* строк, каждая из которых содержит *M* символов. Каждый символ может быть либо «.» (точка), обозначающим поле, либо «W», обозначающим лес, либо «#», обозначающим воду. Гарантируется, что начальная и конечная клетки пути переселенца не являются водой.

**Формат вывода**

В первой строке выходного файла выведите количество единиц времени, необходимое для перемещения поселенца (перемещение в клетку с полем занимает *1* единицу времени, перемещение в клетку с лесом  — *2* единицы времени). Во второй строке выходного файла выведите последовательность символов, задающих маршрут переселенца. Каждый символ должен быть одним из четырех следующих: «N» (движение вверх), «E» (движение вправо), «S» (движение вниз), «W» (движение влево). Если таких маршрутов несколько, выведите любой из них.

Если дойти из начальной клетки в конечную невозможно, выведите число *-1*.

### ***Задача №14 (Свинки-Копилки)***

У Васи есть *n* свинок-копилок, свинки занумерованы числами от *1* до *n*. Каждая копилка может быть открыта единственным соответствующим ей ключом или разбита.

Вася положил ключи в некоторые из копилок (он помнит, какой ключ лежит в какой из копилок). Теперь Вася собрался купить машину, а для этого ему нужно достать деньги из всех копилок. При этом он хочет разбить как можно меньшее количество копилок (ведь ему еще нужно копить деньги на квартиру, дачу, вертолет…). Помогите Васе определить, какое минимальное количество копилок нужно разбить.

**Формат ввода**

В первой строке содержится число *n* — количество свинок-копилок *(1 ≤ n ≤ 100)*. Далее идет *n* строк с описанием того, где лежит ключ от какой копилки: в *i*-й из этих строк записан номер копилки, в которой находится ключ от *i*-й копилки.

**Формат вывода**

Выведите единственное число: минимальное количество копилок, которые необходимо разбить.

### ***Задача №15 (Долой списывание!)***

Во время теста Михаил Дмитриевич заметил, что некоторые лкшата обмениваются записками. Сначала он хотел поставить им всем двойки, но в тот день Михаил Дмитриевич был добрым, а потому решил разделить лкшат на две группы: списывающих и дающих списывать, и поставить двойки только первым.

У Михаила Дмитриевича записаны все пары лкшат, обменявшихся записками. Требуется определить, сможет ли он разделить лкшат на две группы так, чтобы любой обмен записками осуществлялся от лкшонка одной группы лкшонку другой группы.

**Формат ввода**

В первой строке находятся два числа *N* и *M* — количество лкшат и количество пар лкшат, обменивающихся записками (, Изображение выглядит как Шрифт, текст, линия, типография

Автоматически созданное описание). Далее в *M* строках расположены описания пар лкшат: два различных числа, соответствующие номерам лкшат, обменивающихся записками (нумерация лкшат идёт с 1). Каждая пара лкшат перечислена не более одного раза.

**Формат вывода**

Необходимо вывести ответ на задачу Павла Олеговича. Если возможно разделить лкшат на две группы, выведите «YES»; иначе выведите «NO».

### ***Задача №16 (Авиаперелёты)***

Главного конструктора Петю попросили разработать новую модель самолёта для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелёта между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолёт сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

**Формат ввода**

Первая строка входного файла содержит натуральное число *n* () — число городов в Бубундии. Далее идут *n* строк по *n* чисел каждая. *j*-е число в *i*-й строке равно расходу топлива при перелёте из *i*-го города в *j*-й. Все числа не меньше нуля и меньше *109*. Гарантируется, что для любого *i* в *i*-й строчке *i*-е число равно нулю.

**Формат вывода**

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

## **Тимус**

### ***1005 (Куча камней)***

У вас есть несколько камней известного веса *w*1, …, *wn*. Напишите программу, которая распределит камни в две кучи так, что разность весов этих двух куч будет минимальной.

**Исходные данные**

Ввод содержит количество камней *n* (1 ≤ *n* ≤ 20) и веса камней *w*1, …, *wn* (1 ≤ *wi* ≤ 100 000) — целые, разделённые пробельными символами.

**Результат**

Ваша программа должна вывести одно число — минимальную разность весов двух куч.

### ***1444 (Накормить Элефпотама)***

Гарри Поттер сдаёт экзамен по предмету «Уход за магическими существами». Его задание — накормить карликового элефпотама. Гарри помнит, что элефпотамы отличаются прямолинейностью и невозмутимостью. Они настолько прямолинейны, что ходят строго по прямой, и настолько невозмутимы, что заставить их идти можно, только если привлечь его внимание к чему-нибудь действительно вкусному. И главное, наткнувшись на цепочку своих собственных следов, элефпотам впадает в ступор и отказывается идти куда-либо. По словам Хагрида, элефпотамы обычно возвращаются домой, идя в обратную сторону по своим собственным следам. Поэтому они никогда не пересекают их, иначе могут заблудиться. Увидев свои следы, элефпотам детально вспоминает все свои перемещения от выхода из дома (поэтому-то они и ходят только по прямой и лишний раз не меняют направление — так легче запоминать). По этой информации элефпотам вычисляет, в какой стороне расположена его нора, после чего поворачивается и идет прямо к ней. Эти вычисления занимают у элефпотама некоторое (довольно большое) время. А то, что некоторые невежды принимают за ступор, на самом деле есть проявление выдающихся вычислительных способностей этого чудесного, хотя и медленно соображающего животного!

Любимое лакомство элефпотамов — слоновьи тыквы, именно они и растут на лужайке, где Гарри должен сдавать экзамен. Перед началом испытания Хагрид притащит животное к одной из тыкв. Скормив элефпотаму очередную тыкву, Гарри может направить его в сторону любой оставшейся тыквы. Чтобы сдать экзамен, надо провести элефпотама по лужайке так, чтобы тот съел как можно больше тыкв до того, как наткнется на свои следы.

**Исходные данные**

В первой строке входа находится число *N* (3 ≤ *N* ≤ 30000) — количество тыкв на лужайке. Тыквы пронумерованы от 1 до *N*, причем номер один присвоен той тыкве, у которой будет стоять элефпотам в начале экзамена. В следующих *N* строках даны координаты всех тыкв по порядку. Все координаты — целые числа от −1000 до 1000. Известно, что положения всех тыкв различны, и не существует прямой, проходящей сразу через все тыквы.

**Результат**

В первой строке выхода вы должны вывести *K* — максимальное количество тыкв, которое может съесть элефпотам. Далее по одному числу в строке выведите *K* чисел — номера тыкв в порядке их обхода. Первым в этой последовательности всегда должно быть число 1.

### ***1450 (Российские газопроводы)***

**Вступление**

Большими неприятностями обернулся прошедший год для Государства Российского. То неурожай, то птичий грипп, то вечные споры хозяйствующих субъектов... A тут ещё и Президент задумал, наконец, собрать средства на покупку новой балалайки и ручного медведя для своего двоюродного племянника. Все эти факторы (в особенности, конечно, последний) сильно ударили по экономике государства. Посовещавшись со своими друзьями в валенках и ушанках, Президент решил воспользоваться традиционным методом укрепления национального бюджета - увеличением налога на транспортировку газа.

**Задача**

Сеть российских газопроводов представляет собой N перекачивающих станций, некоторые из которых соединены газопроводами. Для каждого из M газопроводов известны номера станций A[i] и B[i], которые он соединяет, и его прибыльность C[i], т.е. то количество долларов, которое будет ежесуточно приносить в виде налогов перекачка газа по этому газопроводу. Каждая пара станций соединена не более чем одним газопроводом.

Сеть была построена советскими инженерами, которые точно знали, что газ поставляется из месторождений Украины в Сибирь, а не наоборот. Поэтому все газопроводы являются однонаправленными, т.е. для каждого газопровода перекачка газа возможна только в направлении из станции с номером A[i] на станцию с номером B[i]. Более того, для любых двух станций X и Y верно, что если возможна перекачка газа из X на Y (возможно, через промежуточные станции), то обратная перекачка из Y на X невозможна. Известно, что газ поступает на начальную станцию с номером S и отгружается потребителям на конечной станции с номером F.

Президент потребовал от Правительства указать маршрут (т.е. линейную последовательность попарно соединённых газопроводами станций) перекачки газа из начальной станции на конечную, причём прибыльность этого маршрута должна быть максимальной. Под прибыльностью маршрута понимается суммарная прибыльность входящих в него газопроводов.

К сожалению, Президент не учёл того факта, что многие газопроводы изначальной сети уже давно прекратили существование, в результате чего может оказаться, что перекачка газа из начальной станции на конечную вообще невозможна...

**Исходные данные**

Первая строка содержит целые числа N (2 ≤ N ≤ 500) и M (0 ≤ M ≤ 124750). Каждая из следующих M строк содержит целые числа A[i], B[i] (1 ≤ A[i], B[i] ≤ N) и C[i] (1 ≤ C[i] ≤ 10000) для соответствующего газопровода. Последняя строка содержит целые числа S и F (1 ≤ S, F ≤ N; S ≠ F).

**Результат**

Если искомый маршрут существует, выведите его прибыльность. Иначе выведите "No solution".

### ***1628 (Белые полосы)***

У каждого неудачника в жизни бывают не только чёрные, но и белые полосы. Марсианин Вась-Вась отмечает в календаре, представляющем собой таблицу *m* × *n*, те дни, когда ему ужасно не повезло. Если Вась-Васю не повезло в *j*-й день *i*-й недели, то он закрашивает ячейку таблицы (*i*, *j*) в чёрный цвет. Все незакрашенные ячейки в таблице имеют белый цвет.

Будем называть отрезками жизни прямоугольники размером 1 × *l* либо *l* × 1. Белыми полосами Вась-Вась считает все максимальные по включению белые отрезки таблицы. А сможете ли Вы определить, сколько всего белых полос было в жизни Вась-Вася?

**Исходные данные**

Первая строка содержит целые числа *m*, *n*, *k* — размеры календаря и количество неудачных дней в жизни Вась-Вася (1 ≤ *m*, *n* ≤ 30000; 0 ≤ *k* ≤ 60000). В следующих *k* строках перечислены неудачные дни в виде пар (*xi*, *yi*), где *xi* — номер недели, к которой относится неудачный день, а *yi* — номер дня в этой неделе (1 ≤ *xi* ≤ *m*; 1 ≤ *yi* ≤ *n*). Описание каждого неудачного дня встречается только один раз.

**Результат**

Выведите число белых полос в жизни Вась-Вася.

### ***1650 (Миллиардеры)***

Возможно, вы знаете, что из всех городов мира больше всего миллиардеров живёт в Москве. Но, поскольку работа миллиардера подразумевает частые перемещения по всему свету, в определённые дни какой-то другой город может занимать первую строчку в таком рейтинге. Ваши приятели из ФСБ, ФБР, MI5 и Шин Бет скинули вам списки перемещений всех миллиардеров за последнее время. Ваш работодатель просит посчитать, сколько дней в течение этого периода каждый из городов мира был первым по общей сумме денег миллиардеров, находящихся в нём.

**Исходные данные**

В первой строке записано целое число *n* — количество миллиардеров (1 ≤ *n* ≤ 10000). В каждой из следующих *n* строк записаны данные на определённого человека: его имя, название города, где он находился в первый день данного периода, и размер состояния. В следующей строке записаны целые числа *m* и *k* — количество дней, о которых есть данные, и количество зарегистрированных перемещений миллиардеров соответственно (1 ≤ *m* ≤ 50000; 0 ≤ *k* ≤ 50000). В следующих *k* строках записан список перемещений в формате: номер дня (от 1 до *m* − 1), имя человека, название города назначения. Вы можете считать, что миллиардеры путешествуют не чаще одного раза в день и что они отбывают поздно вечером и прибывают в город назначения рано утром следующего дня. Список упорядочен по возрастанию номера дня. Все имена и названия городов состоят не более чем из 20 латинских букв, регистр букв имеет значение. Состояния миллиардеров лежат в пределах от 1 до 100 миллиардов.

**Результат**

В каждой строке должно содержаться название города и, через пробел, количество дней, в течение которых этот город лидировал по общему состоянию миллиардеров, находящихся в нём. Если таких дней не было, пропустите этот город. Города должны быть отсортированы по алфавиту (используйте обычный порядок символов: ABC...Zabc...z).

### ***1726 (Кто ходит в гости…)***

Программный комитет школьных соревнований по программированию, проходящих в УрГУ — многочисленная, весёлая и дружная команда. Дружная настолько, что общения в университете им явно не хватает, поэтому они часто ходят друг к другу в гости. Все ребята в программном комитете очень спортивные и ходят только пешком.

Однажды хранитель традиций олимпиадного движения УрГУ подумал, что на пешие прогулки от дома к дому члены программного комитета тратят слишком много времени, которое могли бы вместо этого потратить на придумывание и подготовку задач. Чтобы доказать это, он решил посчитать, какое расстояние в среднем преодолевают члены комитета, когда ходят друг к другу в гости. Хранитель традиций достал карту Екатеринбурга, нашёл на ней дома всех членов программного комитета и выписал их координаты. Но координат оказалось так много, что хранитель не смог справиться с этой задачей самостоятельно и попросил вас помочь ему.

Город Екатеринбург представляет собой прямоугольник со сторонами, ориентированными по сторонам света. Все улицы города идут строго с запада на восток или с севера на юг, проходя через весь город от края до края. Дома всех членов программного комитета расположены строго на пересечении каких-то двух перпендикулярных улиц. Известно, что все члены комитета ходят только по улицам, поскольку идти по тротуару гораздо приятнее, чем по дворовым тропинкам. И, конечно, при переходе от дома к дому они всегда выбирают кратчайший путь. Программный комитет очень дружный, и все его члены ходят в гости ко всем одинаково часто.

**Исходные данные**

Первая строка содержит целое число *n* — количество членов программного комитета (2 ≤ *n* ≤ 105). В *i*-й из следующих *n* строк через пробел записаны целые числа *xi*, *yi* — координаты дома *i*-го члена программного комитета (1 ≤ *xi*, *yi* ≤ 106).

**Результат**

Выведите среднее расстояние, которое проходит член программного комитета от своего дома до дома своего товарища, округлённое вниз до целых.