

Задача А. Счастливые билеты 2008

Имя входного файла: antipal.in
Имя выходного файла: antipal.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Екатеринбургских трамваях ввели новые билеты. Теперь в их номерах больше нет ведущих нулей, они просто идут по порядку, начиная с единицы. Некоторых пассажиров эта новость привела в замешательство — еще бы, а как теперь определять, счастливый билет или нет?! Новые версии критерия обсуждались в салонах трамваев, на остановках и, конечно, на трамвайном форуме в интернете. Одна из версий была довольно изощренной — счастливым билетом считается такой, в номере которого никакие две противоположные цифры не совпадают, то есть $d_i \neq d_{n+1-i}$ при $i \neq n+1-i$, где $d_1d_2\dots d_n$ — номер билета.

Один из заядлых посетителей трамвайного форума был программистом. Он тут же задался вопросом: а не много ли будет в этом случае счастливых билетов? И решил посчитать, сколько таких билетов в рулоне у кондуктора, если номер первого билета в рулоне — a , а последнего — b , и номера билетов в рулоне идут по порядку.

Формат входного файла

В единственной строке через пробел записаны целые числа a и b ($1 \leq a \leq b \leq 10^{18}$).

Формат выходного файла

Выведите количество счастливых билетов в рулоне.

Примеры

antipal.in	antipal.out
1 10	10
100 2000	1620

Задача В. Трамвайная плитка

Имя входного файла: `bricks.in`
Имя выходного файла: `bricks.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Екатеринбурге активно развивается общественный транспорт. В ближайшее время городские власти планируют отремонтировать около 20 км трамвайных путей по новой технологии с применением специально разработанной плитки. Внедрение нового способа ремонта межрельсового пространства началось минувшим летом, и первые результаты видны уже сейчас.

Так, бригада Равшана и Джамшуда оснастила плиткой почти 2 перекрестка в центре города. Такая поразительная скорость работы объясняется новой хитростью, придуманной их начальником. Перед началом ремонтных работ он сам размещает между рельсами несколько плиток так, чтобы можно было продолжить выкладывать их только одним способом. В результате экономится время на проектирование плиточных узоров, которое занимает большую часть рабочего дня у других бригад.

Каждая плитка имеет размер 1×2 . Требуется замостить плиткой прямоугольную область размером $n \times m$. Определите минимальное число плиток, которое должен выложить начальник, чтобы итоговый узор (то есть полное замощение прямоугольника плитками) определялся однозначно. Два узора считаются одинаковыми, если расположение всех плиток в них совпадает.

Формат входного файла

Единственная строка содержит целые числа n и m ($1 \leq n, m \leq 10$; произведение nm четное).

Формат выходного файла

Выведите в первой строке минимальное число плиток, которое должен выложить начальник. Следующие n строк должны содержать по m символов, описывающих одно из возможных начальных расположений. Символ «1» обозначает клетку, занятую плиткой, символ «0» — пустое место.

Примеры

<code>bricks.in</code>	<code>bricks.out</code>
4 4	2 1100 0110 0000 0000
2 5	2 11110 00000

Задача С. Нынче в моде кактусы

Имя входного файла: `cactus.in`
Имя выходного файла: `cactus.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

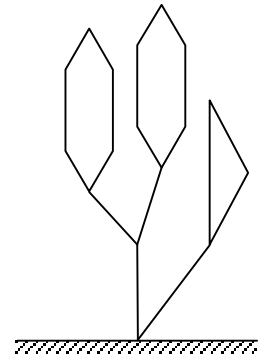
Бесспорно, екатеринбургские трамваи — лучшие в мире. Тем не менее, крупнейшей трамвайной сетью в России обладает Санкт-Петербург. Ещё совсем недавно Санкт-Петербургская трамвайная система была включена в книгу рекордов Гиннеса как самая большая в мире.

Два завсегдатая трамвайного форума из Екатеринбурга решили съездить в Санкт-Петербург на празднование столетнего юбилея пуска трамвая в этом городе. От своих питерских друзей гости узнали, что в течение последних 15 лет объёмы трамвайного движения постоянно сокращаются, на многих магистралях трамвайные пути демонтированы. Так, в центре города трамвайное движение сведено к минимуму, а пассажирская сеть разделена на 3 фрагмента, так что из любой части Санкт-Петербурга уже нельзя добраться в любую другую на трамвае.

А ещё екатеринбуржцы узнали, что в Санкт-Петербурге теперь в моде кактусы. По возвращении в Екатеринбург они решили посадить в своём рабочем кабинете кактус. Кактус — это связный неориентированный граф, каждое ребро которого лежит не более чем на одном простом цикле. Одна вершина кактуса соприкасается с землёй и называется корнем кактуса.

Но вскоре выяснилось, что кактусы стали слишком популярны и уже растут дома у всех заядлых посетителей трамвайного форума. Тогда друзья решили избавиться от кактуса весьма оригинальным способом. Они по очереди выбирают некоторое ребро кактуса и разрубают его. При этом оно удаляется, и если кактус разбивается на две части, то та часть, которая больше не связана с корнем, выбрасывается.

Друзья договорились, что тот, кто отрубит последнее растущее от корня ребро, получит трамвайный проездной на следующий месяц. Определите, кто из них выиграет при оптимальной стратегии обоих.



Формат входного файла

Вместе с модой на кактусы екатеринбуржцы следуют питерской моде описывать множество рёбер кактуса в виде совокупности рёберно-непересекающихся путей. В первой строке даны целые положительные числа n , m и r , где n — количество вершин кактуса, m — количество путей, r — номер корневой вершины кактуса, $1 \leq r \leq n \leq 50\,000$.

Каждая из следующих m строк содержит путь в виде списка вершин. Путь начинается с числа n_i — длины списка, $2 \leq n_i \leq 100\,000$. Далее в строке следует n_i целых чисел — номера вершин пути в порядке обхода. Смежные вершины любого пути различны. Между каждой парой вершин может лежать только одно ребро; каждое ребро кактуса может встречаться во входных данных только один раз.

Формат выходного файла

Выведите «First», если при правильной игре проездной выиграет тот, кто сделает первый ход. В противном случае выведите «Second».

Примеры

<code>cactus.in</code>	<code>cactus.out</code>
17 2 1 15 3 4 5 6 7 8 3 2 9 10 11 12 13 14 9 6 2 1 15 16 17 15	First
16 2 1 15 3 4 5 6 7 8 3 2 9 10 11 12 13 14 9 5 2 1 15 16 1	Second

Задача D. Децимация

Имя входного файла: `decimation.in`
Имя выходного файла: `decimation.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Децимация — казнь каждого десятого по жребью, высшая мера дисциплинарных наказаний в древнеримской армии.

Вы думаете, легко работать кондуктором в трамвае? Заядлые безбилетники так и норовят проехать бесплатно. А контролёры безжалостно штрафуют не только безбилетников, но и самих кондукторов — за то, что те не справляются со своими обязанностями.

В ходе операции «Безбилетник-2008», которую недавно провела Ассоциация контролёров города Екатеринбурга, выяснилось, что более чем в половине трамваев в момент проверки ехал хотя бы один безбилетник. Главный Контролёр Екатеринбурга рассвирепел и решил прибегнуть к наказанию: выстроить провинившихся кондукторов в колонну и каждого десятого тут же оштрафовать в размере средней месячной зарплаты кондуктора.

Главному Безбилетнику Екатеринбурга стало жалко бедных кондукторов, и он решил им помочь, так как знал, что среди них есть хорошие, вполне справляющиеся со своими обязанностями. Просто некоторые их пассажиры в момент проверки еще не успели оплатить проезд. Перед тем как кондукторов начнут штрафовать, Главный Безбилетник может поставить в колонну некоторых своих друзей-безбилетников. При этом Главный Контролёр, конечно, не догадывается о том, что не все стоящие в колонне — кондуктора. Он оштрафует каждого, чей порядковый номер делится на 10 (люди в колонне нумеруются начиная с единицы). Каким образом Главный Безбилетник должен расставить своих друзей, чтобы в сумме было оштрафовано как можно меньше безбилетников и хороших кондукторов?

Формат входного файла

В первой строке через пробел записано два целых числа: n — количество провинившихся кондукторов ($1 \leq n \leq 10\,000$), и k — количество друзей Главного Безбилетника, готовых помочь кондукторам ($0 \leq k \leq 50$). Вторая строка состоит из n символов: i -й символ равен «1», если изначально на i -м месте в колонне стоит хороший кондуктор, и «0» — если плохой.

Формат выходного файла

В первой строке выведите минимальное суммарное количество оштрафованных безбилетников и хороших кондукторов. Во второй строке выведите число m — сколько безбилетников следует поставить в колонну, и затем m целых чисел — их номера в получившейся колонне. Все числа в строке должны быть разделены пробелами.

Примеры

<code>decimation.in</code>	<code>decimation.out</code>
10 2 0000000001	0 2 5 12
10 2 1111111111	1 0

Задача Е. Трамвайный форум

Имя входного файла: `forum.in`
Имя выходного файла: `forum.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На трамвайном форуме общаются водители трамваев, троллейбусов и автобусов. Мы прочитали все сообщения на форуме за последний месяц и поняли, что те пользователи, в чьих сообщениях слово «tram» встречается чаще, чем слово «trolleybus» — это водители трамваев. Аналогично, те пользователи, в чьих сообщениях слово «trolleybus» встречается чаще, чем слово «tram» — это водители троллейбусов. Все остальные пользователи трамвайного форума — водители автобусов. Сможете ли вы, зная это, определить, кто оставил на трамвайном форуме последнее сообщение?

Формат входного файла

Входные данные содержат последнее сообщение на трамвайном форуме. Сообщение состоит из латинских букв в нижнем регистре, пробелов, переводов строк и знаков препинания: точек, запятых, тире, двоеточий, восклицательных и вопросительных знаков. Общая длина сообщения не превосходит 10 000 символов. Слова состоят из букв и разделяются пробелами, переводами строк и знаками препинания.

Формат выходного файла

Выведите «Tram driver», если пользователь, оставивший последнее сообщение, — водитель трамвая, «Trolleybus driver», если он — водитель троллейбуса, и «Bus driver», если он — водитель автобуса.

Примеры

<code>forum.in</code>	<code>forum.out</code>
<code>of course, tram is the best transport in the world!</code>	<code>Tram driver</code>
<code>yes, yekaterinburg trams are the best trams in the world, but i like my trolleybus too.</code>	<code>Trolleybus driver</code>
<code>read forum.tr.ru every day! you can learn much about tram drivers and trolleybus drivers.</code>	<code>Bus driver</code>

Задача F. Для любителей статистики

Имя входного файла: `queries.in`
Имя выходного файла: `queries.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вы никогда не задумывались над тем, сколько человек за год перевозят трамваи города с десятиллионным населением, в котором каждый третий житель пользуется трамваем по два раза в день?

Предположим, что на планете Земля n городов, в которых есть трамваи. Любители статистики подсчитали для каждого из этих городов, сколько человек перевезено трамваями этого города за последний год. Из этих данных была составлена таблица, в которой города были отсортированы по алфавиту. Позже выяснилось, что для статистики названия городов несущественны, и тогда их просто заменили числами от 1 до n . Поисковая система, работающая с этими данными, должна уметь быстро отвечать на вопрос, есть ли среди городов с номерами от l до r такой, что за год трамваи этого города перевезли ровно x человек. Вам предстоит реализовать этот модуль системы.

Формат входного файла

В первой строке дано целое число n , $0 < n < 70\,000$. В следующей строке приведены статистические данные в виде списка целых чисел через пробел, i -е число в этом списке — количество человек, перевезенных за год трамваями i -го города. Все числа в списке положительны и не превосходят $10^9 - 1$. В третьей строке дано количество запросов q , $0 < q < 70\,000$. В следующих q строках перечислены запросы. Каждый запрос — это тройка целых чисел l , r и x , записанных через пробел; $1 \leq l \leq r \leq n$; $0 < x < 10^9$.

Формат выходного файла

Выведите строку длины q , в которой i -й символ равен «1», если ответ на i -й запрос утвердителен, и «0» в противном случае.

Пример

<code>queries.in</code>	<code>queries.out</code>
5 1234567 666666 3141593 666666 4343434	10101
5 1 5 3141593 1 5 578202 2 4 666666 4 4 7135610 1 1 1234567	

Задача G. Нацпроект «Трамваи»

Имя входного файла: `routes.in`
Имя выходного файла: `routes.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Президент объявил развитие трамвайного движения приоритетным национальным проектом. Бюджетных средств, выделенных Екатеринбургу в рамках этого проекта, должно хватить на полную реконструкцию трамвайной сети города.

В Екатеринбурге $2n$ трамвайных остановок. В ходе реконструкции решено оставить все остановки на своих местах и не строить новых. Пути же должны быть проложены так, чтобы от любой остановки до любой другой можно было доехать на трамвае без промежуточных остановок.

После изучения сообщений, оставленных на трамвайном форуме, выяснилось, что горожане будут довольны лишь в том случае, если после окончания реконструкции для любой пары остановок будет существовать трамвайный маршрут, на котором можно добраться от одной из этих остановок до другой, не останавливаясь на промежуточных остановках. Понятно, что этому требованию удовлетворяет сеть из $n(2n-1)$ маршрутов, состоящих всего из двух остановок каждый. Но мэр Екатеринбурга заинтересован в том, чтобы после реконструкции осталось ровно n трамвайных маршрутов, совершающих по пути следования ровно $2n$ остановок каждый (включая конечные). Трамваи должны двигаться по этим маршрутам в обоих направлениях. Как провести реконструкцию так, чтобы и мэр города, и жители остались довольны?

Формат входного файла

В единственной строке дано целое число n , $1 \leq n \leq 100$.

Формат выходного файла

Выведите n строк, содержащих описание трамвайных маршрутов. Каждый маршрут — последовательность целых чисел от 1 до $2n$, записанных через пробел. Маршрут может проходить несколько раз через одну остановку. Если задача имеет несколько решений, можно вывести любое из них. Если задача не имеет решения, выведите в единственной строке число -1 .

Пример

<code>routes.in</code>	<code>routes.out</code>
3	1 6 2 1 3 4 2 3 6 5 4 6 5 1 4 2 5 3

Задача Н. Трамвайный пасьянс

Имя входного файла: `solitaire.in`
Имя выходного файла: `solitaire.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Казалось бы, пробки на дорогах — это не проблема для трамваев. Но если пути проходят прямо по проезжей части, то трамвай оказывается на равных с другим транспортом. На самом деле, ему даже хуже — ведь трамвай не может объехать проблемный участок дороги.

Водитель трамвая Зина всегда найдет, чем заняться, пока ее вагон стоит в пробке. Больше всего она любит раскладывать один карточный пасьянс. В нем используется колода, состоящая из карт двух мастей (червей и пик) n различных достоинств (то есть всего $2n$ карт). Карты берутся из колоды по одной и выкладываются в ряд слева направо рубашкой вниз. После выкладывания каждой карты производится *просеивание*: если в ряду есть две карты одной масти либо одного достоинства, расположенные через одну, то левая из этих двух карт удаляется, а все карты правее нее сдвигаются на одну влево. При этом если таких пар несколько, то выбирается та, которая находится ближе к началу ряда. Просеивание производится многократно, пока в ряду есть такие пары. Пасьянс сошелся, если после выкладывания всей колоды и просеивания в ряду осталось ровно две карты.

Например, если в определенный момент в ряду были карты «H7 S5 S2 H4» («H» обозначает черви, «S» — пики), и к ним добавилась «S7», то после первого просеивания получится «H7 S5 H4 S7». Далее из двух пар «просеивается» первая, и остается «S5 H4 S7». И, наконец, после очередного просеивания получаем «H4 S7».

Зине всегда было интересно, каков шанс, что пасьянс сойдется. Сможете ли вы ответить на этот вопрос?

Формат входного файла

В единственной строке записано целое число n , $2 \leq n \leq 10000$.

Формат выходного файла

Выведите вероятность того, что пасьянс сойдется, в виде несократимой дроби. Используйте формат, приведенный в примере.

Пример

<code>solitaire.in</code>	<code>solitaire.out</code>
2	2/3

Задача I. Квадратная страна 4

Имя входного файла: `square.in`
Имя выходного файла: `square.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Квадратной стране никогда не было трамваев. Конечно, это не могло не волновать население. И вот на только что проведенном референдуме народ высказался за строительство трамвайной сети по всей стране, а также за то, чтобы эта сеть была связана с трамвайной сетью соседней Прямоугольной страны. Народ проголосовал — значит надо строить. Но уже на стадии проектирования возникла проблема, связанная с тем, что системы координат Квадратной и Прямоугольной стран различаются, более того, у них даже координатные оси не параллельны.

Президент Квадратной страны посоветовался с Квадратной Думой и принял историческое решение: повернуть систему координат своей страны относительно точки $(0, 0)$ на угол α по часовой стрелке. Решение оказалось весьма непопулярным — ведь в Квадратной стране все участки, находящиеся в частной собственности, представляют собой наборы квадратов со сторонами, параллельными осям, и вершинами с целочисленными координатами. Выходит, что после поворота придется изменить границы частных владений. Порядок установки новых границ был утвержден указом Президента. Собственник заново определяется для каждой клетки единичного размера. Если некоторый собственник владел до поворота более чем половиной этой клетки, то он становится владельцем всей клетки. Если же такого собственника нет, то вся клетка отходит государству.

Квадратное Правительство поручило вам автоматизировать перераспределение частной собственности. Вам дана карта, на которой отмечены все нынешние частные владения. Требуется построить карту, где будут отмечены все частные владения после поворота.

Формат входного файла

Так как земли, расположенные далеко от центра, не пользуются спросом, все частные владения в Квадратной стране расположены внутри квадрата $[-n, n] \times [-n, n]$. Схема владений представляет собой квадратную таблицу, в каждой ячейке которой записан уникальный код владельца соответствующей клетки — строчная латинская буква. Если в ячейке стоит точка, то соответствующая клетка принадлежит государству. На схеме ось x направлена слева направо, а ось y — снизу вверх.

В первой строке через пробел записаны два числа: n и α ; $1 \leq n \leq 30$; $0 \leq \alpha \leq 90$; угол α дан в градусах. В следующих $2n$ строках записано по $2n$ символов, образующих схему владений квадрата $[-n, n] \times [-n, n]$.

Формат выходного файла

После поворота некоторые частные владения уже могут не попасть в квадрат $[-n, n] \times [-n, n]$, поэтому выведите в том же формате схему владений квадрата $[-2n, 2n] \times [-2n, 2n]$ после поворота координат и перераспределения собственности.

Пример

<code>square.in</code>	<code>square.out</code>
2 45.0
aaaa
aaaa	...aa...
bb.x	..aa.x..
bbx.	..a..x..
	...bb...

Задача J. Ползуны

Имя входного файла: `wheels.in`
Имя выходного файла: `wheels.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Екатеринбурге достаточно разветвленная трамвайная сеть. По некоторым улицам ходят трамваи более чем десяти маршрутов. Понятно, что от такой нагрузки рельсы быстро изнашиваются. Около остановок пути вообще иногда имеют волнообразный вид. Многие винят в этом городские власти — ремонт путей производится редко, да и если меняют рельсы, то на самые дешевые: из низкокачественного металла, с деревянными шпалами.

Однако в плохом состоянии путей виноваты не только власти. Оказывается, некоторые водители трамваев — любители быстрой езды — изрядно портят и рельсы, и свои трамваи. Если трамвай разгоняется до большой скорости, скажем, до 80 км/ч, а потом резко тормозит перед остановкой, то он проезжает некоторый участок юзом. Это приводит не только к порче рельсов, но и к стиранию колес в месте их соприкосновения с рельсами. На колесах появляются плоские участки — так называемые ползуны. Их наличие приводит к стуку колес, дребезжанию вагона и еще большей порче рельсов.

Руководство трамвайного депо решило улучшить ситуацию и провести техобслуживание всех трамваев с ползунами на колесах. Для того чтобы ликвидировать ползуны, колесные пары обтачивают на специальном станке. Колеса одной пары при этом уменьшаются до одного размера, но разные пары могут уменьшиться по-разному. Чтобы оснастить колесами один вагон, ремонтники должны выбрать четыре колесные пары (именно столько их у вагона) с одинаковым диаметром.

Ремонтники закончили обточку всех колес. Некоторые из них уменьшились настолько, что стали непригодны для использования. Такие колеса идут на металлолом. Вместо них можно воспользоваться запасом колесных пар, оставшихся от списанных вагонов. Ремонтники измерили все колесные пары, которые можно устанавливать на вагоны, и теперь хотят понять, на сколько вагонов их хватит.

Формат входного файла

В первой строке записано целое число n — количество имеющихся колесных пар ($1 \leq n \leq 150$). В следующих n строках записаны диаметры колес в миллиметрах (целые числа от 600 до 700).

Формат выходного файла

Выведите количество вагонов, которое удастся оснастить колесными парами из имеющегося набора.

Пример

<code>wheels.in</code>	<code>wheels.out</code>
9	1
689	
690	
690	
689	
689	
690	
689	
688	
688	