Задача A. Свои люди (Random Access Memory)

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На экране уже который час высвечивалось уведомление «Пользователь Элис оставил Вам сообщение. Прослушать?» Крис не спешил: чтобы отвечать на голосовые сообщения, ему требовалось особое настроение. Наконец, он откинулся на спинку кресла, привычным движением нажал пару клавиш и прикрыл глаза.

— Здравствуйте, меня зовут Элис. Я представляю фирму «Звезды информации», нам дали очень хорошие рекомендации, у нас есть для Вас интересное предложение! — щебетал высокий женский голос, — Оно касается Ваших студентов. Мы — молодая фирма ...

«Студентов» — это было неожиданно. Крис немного преподавал в местном классическом университете на факультете естественных и точных наук и был совершенно уверен, что знает все фирмы, которые могли бы заинтересоваться выпускниками этого факультета. Он отметил про себя, что название фирмы довольно странное, но стал слушать намного внимательнее.

Элис рассказывала о том, что фирме нужны молодые и амбициозные разработчики программного обеспечения с хорошим образованием, что фирма готова создать отличные условия для студентов. И, наконец, что она, Элис, приглашает Криса в офис — посмотреть прекрасно оборудованные рабочие места и комнаты отдыха. Например, завтра, в половине четвертого пополудни. Однако Элис ни разу не обмолвилась о том, что за программное обеспечение разрабатывает фирма и какие используются средства разработки.

Впрочем, Крис был почти уверен, что фирму возглавляет кто-то из его знакомых. Он взглянул на карту: судя по всему, офис располагался недалеко от дома Криса. К тому же завтра — вторник, а по вторникам после трех он свободен. Еще раз подумав, что это наверняка кто-то из своих, Крис записал сообщение «Хорошо. Буду.» и отправил его Элис.

По дороге домой он продолжил размышлять, кто бы мог оказаться во главе этой еще неизвестной ему фирмы, какое программное обеспечение она разрабатывает и какой язык при разработке является основным.

Крис строит свои предположения в виде троек (s_i, p_i, l_i), где s_i — фамилия одного из его знакомых, p_i — программное обеспечение, которое может разрабатывать фирма, а l_i — язык программирования, на котором может вестись разработка. Однако на каждое предположение он тут же находит опровержение, вспоминая, что один из элементов этой тройки совершенно точно должен быть исключен из рассмотрения. Обратите внимание, что, например, язык программирования Хаскель и человек с фамилией Хаскель — две большие разницы (т.е. два совершенно разных элемента).

Пока Крис формирует очередную тройку, он полностью забывает и предыдущие тройки, и опровержения. Поэтому, даже если он уже опровергал некоторый элемент, он может вновь включать его в последующие тройки.

Крис уже рассмотрел N таких троек и к каждой нашёл опровержение. Вам дана эта последовательность троек с опровержениями. После каждой тройки Вам нужно сообщить, сколько на текущий момент существует еще не опровергнутых Крисом элементов на каждой из трёх позиций. Элемент на некоторой позиции считается не опровергнутым, если Крис уже предполагал, что этот элемент может находиться на этой позиции, но еще не находил опровержения, запрещающего этому элементу находиться на этой позиции.

Формат входного файла

В первой строке содержится единственное целое число N ($1 \le N \le 100$) — количество троек с опровержениями.

В каждой из следующих N строк содержится по одной тройке с опровержением. Каждый элемент тройки представляет собой непустую строку длиной не более 50 символов, содержащую строчные латинские буквы. Все элементы тройки различны и отделяются друг от друга одним пробелом.

Затем через пробел записано опровержение, совпадающее с одним из элементов тройки.

Формат выходного файла

Выходной файл состоит из N строк.

В каждой строке выведите три целых числа через пробел: сколько в текущий момент существует не опровергнутых Крисом элементов на первой, второй и третьей позициях соответственно.

Примеры

input.txt	output.txt
3 gates windows pascal gates gates windows pascal windows gates windows pascal pascal	0 1 1 0 0 1 0 0 0
2 haskell matrix pascal haskell pascal matrix haskell haskell	0 1 1 1 1 1
3 carmack fallout cpp carmack carmack fallout cpp carmack carmack fallout cpp carmack	0 1 1 0 1 1 0 1 1

Задача B. Время читать

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Крис сидел в кабинете Элис. Элис говорила очень быстро; впрочем, Крис имел привычку слушать собеседника до тех пор, пока он не умолкнет или не попросит ответить на вопрос. Так что перебивать он ее не собирался и лишь отмечал про себя то, что казалось ему важным или хотя бы любопытным.

Он обратил внимание на фразу о том, что руководитель фирмы имеет связи с очень важными людьми, которые работают на рынке приборов по учету коммунального трафика. Запомнил, что фирма планировала снять офис в центре города, но не нашлось подходящего размера («Это был совершенно не вопрос денег! Нам было нужно много просторных помещений, а не крохотных комнатушек!»). Наконец, отметил, что Элис, похоже, любит очень яркую косметику и массивные украшения. Однако он считал это типичным для девушек маленького роста. «И еще она наверняка курит» — подумал Крис.

— Мы хотели бы взять ваших студентов на трехмесячную практику. Человек пятьдесят. Мы из них выберем десять-пятнадцать. Может быть и больше, если они нам понравятся. Нам нужны лучшие из лучших, — Элис замолчала.

Пятьдесят... Да если собрать всех выпускников факультета естественных и точных наук — всех, не только математиков, но и физиков, и химиков, и биологов, — столько не наберется. Странно, если она этого не знает. Крис выдержал небольшую паузу:

— Видите ли, наши выпускники пользуются большим спросом на рынке труда. Возможно, двое или трое могли бы заинтересоваться работой в Вашей фирме, но вряд ли больше.

— В ближайшее время наша фирма получит большой заказ с очень хорошим бюджетом. Кстати, может быть Вы сами заинтересуетесь? Один специалист стоит десятка студентов, — Элис протянула ему толстую стопку листов и, видя замешательство Криса, добавила: — Наш генеральный директор очень хотел встретиться с Вами сегодня, но он немного задерживается.

У Криса есть своя методика чтения больших документов. Будем называть фрагментом документа некоторое количество расположенных подряд листов стопки. Он выбирает какой-либо фрагмент, который его заинтересовал, и прочитывает именно его. Чтобы не путаться, он помечает прочитанные листы маркером какого-то цвета. Затем он выбирает следующий фрагмент, и читает уже его, также помечая листы маркером, но уже другого цвета. Фрагменты могут перекрываться, и в этом случае в очередном выбранном фрагменте Крис прочитает все листы, которые еще не были помечены маркером, а также все листы, которые помечены маркером того цвета, которым помечен первый лист выбранного фрагмента (разумеется, если он помечен; при наличии нескольких пометок на листе Крис считает актуальной последнюю пометку). Для определенности можно считать, что при чтении фрагмента j Крис будет использовать маркер цвета j.

Один лист Крис прочитывает за 1 единицу времени. Ваша задача —по заданной последовательности фрагментов определить, сколько времени потребуется Крису, чтобы прочесть их.

Формат входного файла

В первой строке содержатся целые числа N и Q ($1 \le N, Q \le 3 \cdot 10^5$) — количество листов в документе и количество фрагментов, выбранных Крисом для чтения.

В каждой из следующих Q строк содержатся по два числа L_j и R_j ($1 \le L_j, R_j \le N$) — номера первой и последней страницы очередного фрагмента.

Формат выходного файла

В первой строке выведите целое число — время, которое потребуется Крису на прочтение всех выбранных фрагментов.

Примеры

input.txt	output.txt
15 5 1 6 4 8 2 7 10 12 7 13	20

Задача C. Дорога

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— Все-таки мне кажется неосмотрительным показывать такую документацию совершенно постороннему человеку, — Крис вернул стопку Элис и уже вознамерился завершить разговор и откланяться.

— Ну уж не такой и посторонний! — чуть рокочущий голос раздался из-за спины Криса. Крис обернулся. Да, это был Саймон. Их дороги с Крисом пересекались уже несколько раз. Саймон считал себя великим программистом и еще более великим руководителем. Его ничуть не смущало, что до сих пор все его проекты и фирмы лопались, как мыльные пузыри. В чем, впрочем, никак ему нельзя было отказать — так это в умении столь красочно рассказывать о светлом будущем очередной своей задумки, что многие из тех, кто имел возможность уже не один раз наблюдать бесславное окончание предыдущих планов, начинали верить, что уж на этот раз у Саймона все получится.

— На самом деле мы временно ютимся в этом здании, сейчас для нас проектируют новый офис в поселке «Синий Тракторист». Окна прямо в лес будут выходить! Там, правда, еще придется немного повозиться, чтобы построить нормальную дорогу для спецтехники, а то понастроили заборов. Но у меня уже есть нужные бумаги, так что заборовладельцам придется подвинуться! — Саймону явно не терпелось продемонстрировать свою значительность.

С некоторых пор поселок «Синий Тракторист» облюбовали весьма состоятельные люди, и земля там стала стоить весьма дорого. Крису было ясно, что участок, на котором собрался строить здание Саймон, практически окружен лесом, к нему сложно подвести и дорогу, и коммуникации. Да и владельцы коттеджей явно будут не в восторге от того, что часть их участков может стать частью дороги.

Карта поселка представляет собой прямоугольник размера $N \times M$. Будем считать, что он разбит на квадраты с единичной стороной. Частные землевладения также представляют собой прямоугольники, стороны которых параллельны сторонам карты, при этом для каждого квадрата с единичной стороной верно, что он либо полностью принадлежит некоторому землевладению, либо полностью не принадлежит никакому землевладению. Никакие два землевладения не перекрываются. Квадраты с координатами $(1, 1)$ (левый верхний угол) и (N, M) (правый нижний угол) не заняты никакими землевладениями. Дорога должна соединить эти квадраты.

Каждый отрезок дороги, которую хочет построить Саймон, должен быть параллелен одной из сторон карты. Дорога должна иметь ширину, равную стороне квадрата, и каждый квадрат должен либо полностью принадлежать ей, либо полностью не принадлежать ей. Наконец, дорога **должна быть кратчайшей**.

Понятно, что при строительстве дороги некоторая часть некоторых землевладений будет отчуждена. Назовем ценой вопроса число K — максимальное количество квадратов для одного землевладения, которое будет отчуждено. Ваша задача — определить минимальную цену вопроса, необходимую для построения кратчайшей дороги, а также вывести саму дорогу.

Формат входного файла

В первой строке содержатся целые числа N и M через пробел ($2 \le N \le 1000$, $2 \le M \le 1000$) — размеры карты поселка.

Следующие N строк содержат по M символов каждая и описывают карту. Свободные квадраты отмечены символом «.», квадраты, занятые каким-либо землевладением, отмечены строчными латинскими буквами от «a» до «z». Каждому землевладельцу принадлежит только одна связная прямоугольная область, разные буквы служат лишь для разграничения связных областей, касающихся друг друга.

Формат выходного файла

В первой строке выведите целое число K — минимально возможную цену вопроса при построении

кратчайшей дороги.

В следующих $N + M - 1$ строках выведите в порядке следования от $(1, 1)$ (левого верхнего угла) до (N, M) (правого нижнего угла) координаты квадратов, которые составят дорогу.

Примеры

input.txt	output.txt
4 5 .aa.. .aacc bb.cc bb...	1 1 1 2 1 2 2 3 2 3 3 3 4 4 4 4 5
4 5 .aa.. .aa.. bbcc. bbcc.	2 1 1 1 2 1 3 1 4 1 5 2 5 3 5 4 5

Задача D. Тестовая задача

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— Крис, я знаю, у тебя нет столько людей. А все потому, что цепляетесь за устаревшие методы обучения!

Несколько лет назад Саймон заявился в университет с суперидеей — перевести студентов на дистанционное обучение: «Преподаватель у себя дома, говорит в камеру, а студенты у себя дома, слушают!». Обещал оптовые поставки оборудования и размахивал листами с диаграммами, показывающими, сколько денег сможет получить университет, сдавая высвободившиеся аудитории и лаборатории под офисы. Классический университет уже тогда был одним из немногих, в котором еще сохранилось очное обучение. Саймона тогда, по счастью, прогнали, а вскоре появилась «Ассоциация классических и медицинских университетов», и проблем у университета стало в разы меньше. Не то чтобы они исчезли совсем, но таких, как Саймон, теперь просто не пустят на порог.

— Так вот, Крис, у меня люди есть. Мы уже набрали около ста выпускников Академического университета информационных и сетевых технологий. Но они специализировались на «Проблеме 2038», и их надо бы немного подучить. Короче, Крис, мне нужен ты. Как преподаватель языка программирования $K\pm$. Вот программа, по которой их надо обучить. Первая группа тебя уже ждет, — Саймон никогда не предполагал, что кто-то может отказаться от его предложения.

Крис слишком хорошо знал Саймона. Он пролистал программу: да, в классическом университете это рассказывают первокурсникам. Ничего сложного. Работать с Саймоном, тем не менее, ему категорически не хотелось, но он все еще надеялся избежать открытого конфликта:

— Понимаешь, Саймон, даже самая хорошая программа должна быть адаптирована под аудиторию. Сначала нужно провести тестирование.

— Так проведи! Сейчас Элис проводит тебя в аудиторию.

Спустя несколько минут Крис уже диктовал группе задачу.

Джон получил интересное предложение от своего провайдера коммунального трафика — оплачивать часть услуг, потребленных в определенное время суток, по более выгодному тарифу. Более того, Джон даже может выбрать сам, какой отрезок суток он будет оплачивать по этому тарифу. Требование провайдера состоит в том, что отрезок должен начинаться и заканчиваться в течение одних суток и иметь длительность M долей суток.

Прибор учета, используемый провайдером, имеет шкалу из N долей суток, поэтому начало и окончание выбранного отрезка суток должно приходиться на целые значения этой шкалы.

Джон решил подойти к вопросу серьезно, и на протяжении D дней записывал все показания счетчика по истечении очередной доли суток (таким образом, у него имеется $D \times N$ записей). Теперь он решил, что набрал достаточно данных, и хочет выбрать такие M (идущих подряд) долей суток, на которые приходится наибольший суммарный расход коммунального трафика.

Ваша задача — определить номер первой доли суток выбранного Джоном отрезка и объем трафика, который был потреблен в течение D дней, если учитывать только этот отрезок суток.

Формат входного файла

В первой строке содержатся целые числа D, N, M ($2 \le D \le 1000$, $2 \le N \le 1000$, $1 \le M < N$) — количество дней, в течение которых вел учет Джон, количество делений на шкале прибора и количество долей суток, которые провайдер предлагает оплачивать по выгодному тарифу.

Следующие D строк содержат по N целых неотрицательных чисел каждая — показания счетчика, записанные Джоном. Все числа не превосходят 10^9.

Формат выходного файла

В первой строке выведите два целых числа — номер доли суток (при нумерации с 1), с которой начинается выбранный Джоном отрезок длиной M долей, и количество трафика, потребленного в течение D дней при учете только этого отрезка суток.

Если существует несколько вариантов ответа, выведите любой.

Примеры

input.txt	output.txt
5 6 3 12 4 6 8 11 7 3 14 15 9 2 6 2 7 1 8 2 8 0 6 0 4 20 13 1 7 15 1 2 0	2 105
3 12 4 3 5 0 5 3 0 3 0 5 5 0 3 4 0 7 4 7 0 7 0 4 7 4 0 6 1 0 1 0 6 6 0 1 1 6 0	4 42

Задача E. Практики

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

По рядам пробежал ропот. Один из сотрудников поднялся и громко, с напором сказал:

— Что за глупое задание? Мы профессионалы-практики! Нас предупреждали, что придет университетский профессор, но Вы должны понимать, куда пришли!

Аудитория одобрительно загудела.

— Я тоже профессионал, и тоже практик, — Крис был абсолютно спокоен, — Для начала мне нужно увидеть, как вы используете некоторые конструкции языка. Проще и быстрее сделать это на задаче такого рода. Полагаю, что за полчаса вы справитесь.

Спустя час Крис просматривал написанное. Результаты превзошли все его ожидания. Всё, написанное сотрудниками, менее всего напоминало программы. Было ощущение, что в течение получаса они просто записывали все слова, которые казались им относящимися к языку , притом в произвольном порядке и с грамматическими ошибками. Порой Крису требовалось несколько раз прочесть вслух буквосочетания, прежде чем он мог уловить звучание и понять, какое же слово имелось в виду.

— Кстати, у меня идея! Эти среды программирования жутко сложны для восприятия их нормальным человеком. Слишком много условностей! Какую-нибудь запятую забудешь поставить — и всё, ошибка компиляции! Как будто непонятно, что имелось в виду! — по Саймону было видно, что мысль пришла ему в голову только что и он, по своему обыкновению, считает ее гениальной. — Ты напишешь компилятор! С поддержкой голосового ввода. Ну и прочими интеллектуальными распознавателями или как это там называется. Не говори ничего! Я понимаю, что это не на пару дней работы. Но есть одна вещь, с которой ребятам нужно помочь прямо сейчас! Для начала ты напишешь простенькую программку для упрощения коллективной работы...

В процессе коллективной работы над проектом нередко случается так, что разные разработчики хотят назначить один и тот же идентификатор разным программным сущностям, которые мы далее для простоты будем называть элементами. Саймон хочет, чтобы «простенькая программка» разрешала конфликты при назначении идентификаторов.

Крис написал программу, которая работает следующим образом.

Программа получает список элементов, для каждого из которых указаны две характеристики: *предпочтительный идентификатор* и *изменяемость* (можно ли заменить предпочтительный идентификатор другим или же это запрещено). Элементы разбиваются на группы с одинаковыми предпочтительными идентификаторами. Элементы помещаются в группы в том же порядке, в котором они присутствуют в списке.

Далее приведём псевдокод предложенного Крисом алгоритма:

Пока есть хотя бы одна группа

- Выбрать группу с лексикографически минимальным предпочтительным идентификатором
- Если в ней ровно один элемент, присвоить ему предпочтительный идентификатор
- Иначе пройти по порядку все элементы группы
  - Если текущий элемент группы помечен как неизменяемый, присвоить ему предпочтительный идентификатор
  - Иначе найти минимальное натуральное n такое, что *предпочтительный идентификатор_ n* не используется как идентификатор и не является предпочтительным идентификатором какой-либо группы, и сделать *предпочтительный идентификатор_ n* идентификатором текущего элемента
- Удалить текущую группу из разбиения

Однако автоматически заменять идентификаторы нежелательно. Поэтому пользователю надо предоставить возможность заранее узнать, какой идентификатор может получить тот или иной элемент, а также модифицировать предпочтительный идентификатор и изменяемость любого элемента.

Ваша задача — по заданному списку элементов обработать Q запросов двух видов:

1. По заданному номеру элемента в списке вывести идентификатор, который будет назначен этому элементу;
2. По заданному номеру элемента в списке сопоставить ему новый предпочтительный идентификатор и новое значение изменяемости.

Формат входного файла

В первой строке содержится целое число N ($1 \le N \le 10^5$). Каждая из следующих строк соответствует элементу в списке (вторая — первому, третья — второму и т.д.) и содержит предпочтительный идентификатор и значение изменяемости для этого элемента (через пробел).

В строке, следующей за списком, содержится целое число Q ($1 \le Q \le 10^5$).

В каждой из следующих Q строк содержится по одному запросу.

Запрос вида 1 начинается знаком «?» и состоит из единственного целого числа — номера элемента в списке. Запрос вида 2 начинается знаком «*» и имеет три параметра: номер элемента в списке (целое число), новый предпочтительный идентификатор и новое значение изменяемости. Параметры запроса разделены пробелами.

Предпочтительный идентификатор может состоять из строчных латинских букв, цифр от 0 до 9 и символа нижнего подчеркивания «_». Изменяемость может принимать два значения: 0 — замена предпочтительного идентификатора недопустима и 1 — замена предпочтительного идентификатора допустима.

Размер входного файла не превосходит 1 Мб. Гарантируется, что размер выходного файла не превысит 2 Мб.

Формат выходного файла

В каждой строке выходного файла содержится ответ на очередной запрос вида 1 — идентификатор, который будет назначен указанному в запросе элементу.

Примеры

input.txt	output.txt
5 id 0 id 0 id 1 id_1 1 id 1 5 ? 1 ? 2 ? 3 ? 4 ? 5	id id id_2 id_1 id_3
4 id 1 id 1 id_1 1 id_1 1 10 ? 1 ? 2 ? 3 ? 4 * 3 id_1_1 1 * 4 id_1_2 1 ? 1 ? 2 ? 3 ? 4	id_2 id_3 id_1_1 id_1_2 id_1 id_2 id_1_1 id_1_2

Задача F. Парное программирование

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Саймон вел Криса по своим, как он выражался, «хоромам». Показывал просторные комнаты, в каждой из которых сидело не более пяти сотрудников, открыл дверь в тренажерный зал и похвалился, что внизу еще и бассейн есть. Наконец, распахнул дверь в еще одно помещение со словами: «А здесь наши лучшие программисты!» Четыре человека сидели в креслах, более всего похожих на капсулы, и сосредоточенно, не шевелясь, смотрели на мониторы. Заметив озадаченное выражение лица Криса, Саймон похлопал его по плечу: «Супертехнологии! Им не нужно тратить силы, чтобы нажимать кнопки на клавиатуре. Достаточно просто задержать взгляд на нужном символе на экране!»

— Технология известная, но глаза, наверное, сильно устают, — произнес Крис, подумав про себя «Ну сколько же он на этот раз кредитов набрал...»

Саймон, впрочем, не слушал его:

— Кстати, что-то ты скучный алгоритм по назначению идентификаторов предлагаешь. Мои программисты предложили куда более эффектный!

Упомянутый Саймоном алгоритм можно описать следующим образом.

Согласно политике фирмы, идентификатор должен состоять только из строчных латинских букв. Пусть два программиста пожелали использовать один и тот же идентификатор для обозначения разных программных сущностей.

Каждый из них сначала записывает букву z, после чего на каждом шаге заменяет произвольную букву z в уже написанной строке на этот самый идентификатор.

— Видишь, какие разные строки получились? — Саймон был явно горд придумкой.

Крис посмотрел на написанные строки и задумался — действительно ли эти строки имеют своим предшественником один идентификатор?

Ваша задача — по двум заданным строкам определить, каким был исходный идентификатор, или же определить, что такого идентификатора не существует.

Формат входного файла

В первой строке содержится первая из показанных Крису строк.

Во второй строке содержится вторая из показанных Крису строк.

Каждая из строк непуста и содержит не более 50 символов — строчных латинских букв.

Формат выходного файла

В первой строке выведите исходный идентификатор, который хотели использовать программисты. Если существует несколько вариантов, выведите любой.

Если такого идентификатора не существует, выведите в качестве ответа z.

Примеры

input.txt	output.txt
aaazbbb aazbb	azb
hello world	z
azazz aazzz	azz
aaaaaaaaaaazbzcbzcbzcbzcbzcbzcbzcbzcbzcbzcbzc azbazbazbazbazbazbazbazbazbazbzccccccccccccc	azbzc

Задача G. Комната отдыха

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— О! Я не показал еще тебе комнату отдыха! Идем! — Саймон устремился вниз по лестнице, а Крис проследовал за ним. Ему уже порядком надоела эта «экскурсия», но, как ему казалось, они обошли уже практически все здание.

В комнате горел неяркий свет, стены были задрапированы каким-то темным материалом. Крис остановился на пороге, но Саймон подтолкнул его, и он сделал пару шагов. Дверь за ним закрылась. Несколько минут он стоял, не двигаясь. Наконец, глаза его привыкли к полумраку, и он повернулся к двери.

— Мы уже пробовали открыть, и не только эту. Думаю, вполне очевидно, что пока мы не имели успеха в этом предприятии, — к Крису подошли двое, — Я — Боб, а это — Макс.

— Мы тут уже третью неделю «гостим», — усмехнулся Макс, — Кормят, поят, работать предлагают, хотя не заставляют. Вот только выйти нельзя. Даже с этажа.

Боб и Макс рассказали Крису, что двери комнат на этаже открываются не только личными электронными ключами сотрудников, но и «гостевыми» карточками. Карточки устроены довольно просто: на пластиковую основу нанесены контактные дорожки. Всего таких дорожек может быть 30, но не на всех присутствует напыление: кое-где есть просто очерченный прямоугольник. Когда очередная карточка попадала в руки Бобу или Максу «ввиду производственной необходимости», они запоминали ее «схему» и то, какую дверь она открывала, а потом записывали это.

Выслушав их, Крис предположил, что чем ближе друг к другу расположены двери, тем меньше должно быть отличий между ключами. Однако это предположение нужно проверить. Для начала среди имеющихся у Боба и Макса схем необходимо отыскать две самые «похожие» карточки.

Несколько более формально: каждую схему карточки рассмотрим как натуральное число, записанное в двоичной системе счисления. Критерием «похожести» двух карточек будем считать результат операции XOR между описывающими их числами. Таким образом, Ваша задача — найти пару карточек, для которых результат операции XOR между описывающими их числами минимален. Если таких пар несколько, выведите в качестве ответа любую из них.

Формат входного файла

В первой строке содержится целое число N ($2 \le N \le 3 \cdot 10^5$) — количество карточек.

Во второй строке содержится N натуральных чисел $a_1, a_2, \ldots, a_N$ через пробел, описывающих карточки ($1 \le a_i \le 10^9$).

Формат выходного файла

В первой строке выведите через пробел номера двух карточек, XOR между числами на которых минимален. Если решений несколько, выведите любое из них.

Примеры

input.txt	output.txt
6 4 7 10 5 2 8	1 4
5 1 2 3 4 3	3 5

Задача H. IWorm

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Макс оказался инженером-электронщиком, а Боб — специалистом по компьютерным сетям. Саймон уговорил их приехать из другого города, рассказав о новой модели счетчика коммунального трафика, для которого нужно усовершенствовать программное обеспечение. Однако на деле все оказалось не так. Саймон хотел, чтобы новые счетчики закупались в массовом порядке. Но как убедить потребителей коммунального трафика, что им совершенно срочно необходимо приобрести и установить новую модель? Ответ на этот вопрос для Саймона был совершенно очевиден: нужно, чтобы прежние модели счетчиков перестали работать.

— В общем, Саймон хочет под благовидным предлогом заставить пользователей обновить программное обеспечение, а в обновлении будет такая «закладка», которая заставит счетчики работать неправильно. И Саймон съест рынок! — Макс нацелился и легонько надавил пальцем на планшет. На экране высветилось красное яблоко и вылезающий из него довольный червяк.

Макс играет в следующую игру. Имеется поле размером 7×5 клеток. Клетки первой и последней строки этого поля представляют собой норы, в каждую из которых можно поместить Яблочного Червяка. В каждой из остальных клеток может находиться либо яблоко, либо груша, либо эта клетка может быть пустой.

Когда Яблочный Червяк находится в норе, ему можно указать яблоко, и он вытянется от середины норы к середине клетки, где находится яблоко, чтобы съесть его. Можно считать Яблочного Червяка в этот момент отрезком прямой. Конечно, этот отрезок может проходить через другие клетки (касание угла клетки не учитывается). Если в каких-либо из этих клеток будут яблоки — то Яблочный Червяк съест и их.

Однако есть некоторая проблема: Яблочный Червяк очень не любит груши. Настолько не любит, что если отрезок прямой, соединяющий его нору с указанным ему яблоком, пройдет хотя бы через одну клетку, содержащую грушу, у Яблочного Червяка испортится аппетит, и он не станет есть ни одного яблока.

Ваша задача — по заданному расположению яблок и груш определить, какое максимальное количество яблок может съесть за один раз Яблочный Червяк. Также укажите нору, в которую его следует поместить.

Формат входного файла

Входной файл является описанием игрового поля размером 5×5 (норы не показаны).

Яблоки обозначены латинской буквой A, груши — латинской буквой P, пустые клетки — точкой.

Гарантируется, что на поле есть хотя бы одно яблоко.

Формат выходного файла

В первой строке выведите единственное целое число — максимально возможное количество яблок, которые может съесть Яблочный Червяк.

Далее во второй строке выведите два целых числа через пробел — номер строки (1 для верхней строки или 7 для нижней строки) и номер столбца, в которых должна быть расположена нора, из которой вытягивается Яблочный Червяк.

Далее в третьей строке выведите два целых числа через пробел — номер строки и номер столбца, в которых располагается яблоко, на которое нужно указать.

Примеры

input.txt	output.txt
.A..A .AP.A .PA.A ..APA ..APP	5 1 2 6 3
AAP.. AAAP. APAAP AP.AA AP..P	7 1 1 5 4

Задача I. Матч

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Боб обнаружил в местной локальной сети большой видеоархив матчей между командами, играющими в "Нападение из будущего". В каждом матче участвуют две команды. Матч состоит из игр, в каждой из которых победу одерживает та или иная команда (ничьих нет) и продолжается до тех пор, пока какая-либо из команд не одержит ровно N побед.

Боб поступает следующим образом: он скачивает все игры матча (и, таким образом, ему становится известно, сколько игр было в этом матче). Затем он просматривает игры в том порядке, в котором они состоялись, начиная с первой.

Изначально Боб не знает счёт, с которым закончится матч, и даже не знает, кто победит. Конечно, он мог бы посмотреть описание матча, но тогда ему будет совсем неинтересно смотреть игры. К сожалению, исход игр иногда предсказуем. Боб не будет смотреть игру, если перед её началом может указать, кто победит.

Ваша задача — по описанию матча определить, сколько игр этого матча посмотрит Боб.

Формат входного файла

В первой строке содержатся два целых числа N и M ($1 \le M \le 20$, $1 \le N \le M$) — необходимое количество побед и общее количество игр в матче.

Во второй строке содержится описание матча — строка длины M из нулей и единиц. Ноль обозначает, что первая команда проиграла, а 1 — что первая команда выиграла.

Формат выходного файла

В первой строке выведите единственное целое число — количество игр, которое посмотрит Боб.

Примеры

input.txt	output.txt
3 4 0010	2
3 5 01010	4
5 5 00000	1

Note

Будем называть команду 0 «синие», а команду 1 «зелёные».

В первом примере Боб смотрит первые две игры, в них побеждают синие. Счёт 2-0. Поскольку следующая игра не заканчивает матч, в ней побеждают зелёные. Счёт 2-1 и осталась одна игра, значит, в четвёртой игре победят опять синие. В итоге он посмотрел игры 1 и 2.

Во втором примере после трёх игр счёт 2-1, значит, в следующей игре побеждают зелёные, а пятая игра остаётся непредсказуемой. В итоге Боб смотрит игры 1, 2, 3, 5.

В третьем примере одна из команд всухую разгромила другую, и достаточно посмотреть первую игру, чтобы понять, какая.

Задача J. Новый протокол

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пока Саймон пространно излагал свою идею по повышению среднего IQ отделов фирмы, Крис размышлял. Если кому-то нужно было бы просто продать крупную партию счетчиков, то достаточно было бы устроить «локальную аварию». Не исключено, что Саймон, как обычно, стремится к мировому господству, но у его покровителей могут быть планы поскромнее.

— Знаешь, Саймон, у меня, кажется, есть интересное предложение по счетчикам. Я пообщался со своими сока... соседями по комнате. Все можно сделать проще. Вместо новых счетчиков можно выпустить маленькое устройство, которое будет шифровать трафик в целях безопасности. Конечно, шифровать оно ничего не будет. Просто будет передавать данные в немного измененном порядке. Потребители будут довольны — не надо менять счетчик, достаточно купить маленькую коробочку.

Саймон пристально смотрел на Криса с большого экрана, и в его взгляде были и подозрение, и тот блеск, который возникал, когда он полагал, что не упустит своего. Идея ему нравилась, но он спешно пытался сообразить, есть ли в ней подвох.

Спустя несколько минут Крис сидел в кресле в кабинете Саймона и рисовал схему, поясняя ее. Допустим, первые 4 байта — дата, следующие 4 байта — количество потребленной электроэнергии, следующие 4 — количество холодной воды... Счетчик выдает информацию именно в этом порядке. Но программа, которая обрабатывает данные, может нуждаться в другой последовательности байтов для их правильной интерпретации. Конечно, нет ничего сложного в том, чтобы переставить байты непосредственно в программе. Но можно сделать и по-другому...

По мнению Саймона, перепрограммировать устройство обязательно должен сотрудник со специальным ключом. Однако в этом случае возникает проблема: нельзя перевести всю систему одномоментно на новую версию протокола.

В штате обслуживающей компании есть N техников, каждый из которых обслуживает свой участок. Однако по разным причинам техники работают с разной скоростью, и поэтому каждому из них требуется разное количество дней, чтобы посетить всех домовладельцев своего участка и выполнить перепрограммирование устройства. Известно, что в течение дня каждый техник посещает хотя бы одного домовладельца.

Техники работают, не покладая рук. Как только техник перепрограммировал все устройства на своем участке таким образом, чтобы они поддерживали версию протокола V, он не отправляется отдыхать. Нет, на следующий день он вновь пойдет по своему участку, чтобы перепрограммировать устройства, чтобы они поддерживали версию протокола $V + 1$.

Контракт обслуживающей компании заключен на M дней. Изначально все устройства настроены одинаково и поддерживают версию протокола 0.

Ваша задача — определить, какое максимальное количество версий протокола одновременно придется поддерживать при обработке данных.

Формат входного файла

В первой строке содержатся два целых числа N и M ($2 \le N \le 10^5$, $2 \le M \le 10^5$) — количество техников в штате компании и количество дней, на которые заключен контракт.

Во второй строке содержатся N целых чисел D_1, D_2, ..., D_N, где D_j ($1 \le D_j \le 10^5$, $j = 1, 2, \ldots, N$) — количество дней, требующихся технику j, чтобы полностью обойти свой участок.

Формат выходного файла

В первой строке выведите целое число — максимально возможное количество версий протокола, которое придется одновременно поддерживать при обработке данных.

Примеры

input.txt	output.txt
2 30 1 7	3
2 22 3 5	4
5 42 7 2 8 4 2	7

Note

Пояснение к примерам.

В первом примере количество версий протокола, которые придется поддерживать в течение указанного количества дней, будет колебаться от 2 до 3.

Первый техник на своем участке в первый же день поменял на своем участке протоколы во всех устройствах, и, таким образом, к концу первого дня на его участке будет действовать только версия протокола 1.

Второй техник успел обойти только часть своего участка, и теперь у него на участке действуют версии протокола 0 и 1.

В первый день, как можно видеть, достаточно поддерживать две разные версии протокола.

На второй день первый техник обойдет свой участок еще раз и обновит версию протокола до 2 во всех устройствах. Второй же техник продолжает обход своего участка, на котором по-прежнему будут действовать протоколы версии 0 и версии 1. Это значит, что во второй день потребуется поддержка трех версий протоколов одновременно.

На третий день будет необходимо вновь поддерживать три версии протокола, но это уже будут версии 0, 1 и 3.

Заметим, что на седьмой день будет необходима поддержка только двух версий протокола: 1 и 7 (поскольку второй техник завершит обход своего участка).

Во втором примере максимальное количество версий протокола, которое потребуется поддерживать одновременно, будет достигнуто, например, в 13 день.

Задача K. Контроль качества

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— Вообще-то одну полезную вещь мы для него сделали. В день приезда, пока еще не сообразили, что к чему, — меланхолично заметил Макс. – Кое-что для контроля качества. Даже не знаю, где он выкопал такие дисплеи для счетчиков.

Каждая цифра на дисплее отображается с помощью блока из семи элементов. Для определенности занумеруем их так, как показано на рисунке 1.

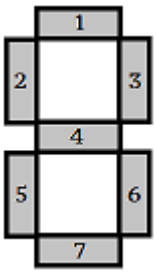
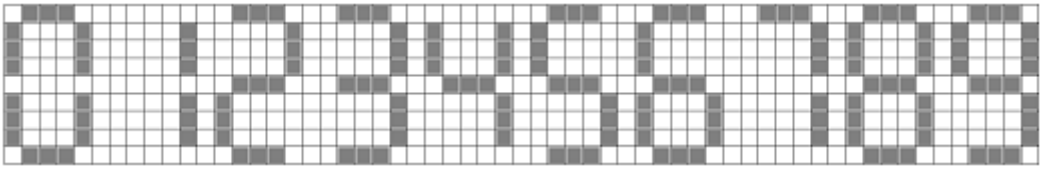


Рисунок 2 демонстрирует, каким образом из этих элементов образуются цифры.



Однако дисплеи, которые Саймон планировал устанавливать в новые счётчики, оказались сделанными не слишком качественно: во многих блоках попадались неисправные элементы. Неисправности были двух видов:

- A — элемент всегда включен;
- B — элемент всегда выключен.

Понятно, что в результате этих неисправностей цифры на дисплее могут стать нечитаемыми. Что хуже, вместо одних цифр могут отобразиться другие: например, всегда выключенный элемент 5 превращает цифру 6 в 5, а цифру 8 — в 9, в то время как всегда включенный элемент 4 превращает 0 в 8.

Впрочем, Саймона это обстоятельство расстраивало мало: он полагал, что «такую мелочь легко перепаять, не заказывать же новые дисплеи!» Но для этого нужно было знать, какие именно элементы в блоке неисправны и каков характер этой неисправности.

Процесс выявления неисправности происходил следующим образом. На блок подавались по одному разу все цифры от 0 до 9, а получаемые изображения цифр фотографировались. Сотрудник, который это делал, нумеровал фотографии соответствующим образом. После этого другой сотрудник рассматривал все 10 фотографий и определял, какие элементы блока являются неисправными.

Макс и Боб частично автоматизировали процесс, придумав, как можно выводить изображения цифр в файл. Но ввод цифр приходилось осуществлять вручную. Тем не менее, Саймон счёл, что теперь двух опытных сотрудников можно заменить одним стажёром. Стажёр отнёсся к делу с энтузиазмом, и спустя пару часов у него уже было несколько тысяч «протестированных» блоков. И одна маленькая проблема — каждый раз он вводил цифры в произвольном порядке.

Ваша задача — написать программу, которая по изображению десяти цифр определит для каждого элемента блока, исправен ли он, а также характер неисправности.

Формат входного файла

Входной файл содержит символьное изображение всех десяти цифр в случайном порядке. Каждое изображение цифры задается матрицей из 9 строк и 5 столбцов, при этом каждый элемент блока (как горизонтальный, так и вертикальный) состоит из трех латинских символов «x», а остальные места заполнены символами «.».

Цифры отделяются друг от друга одним столбцом символов «.».

Формат выходного файла

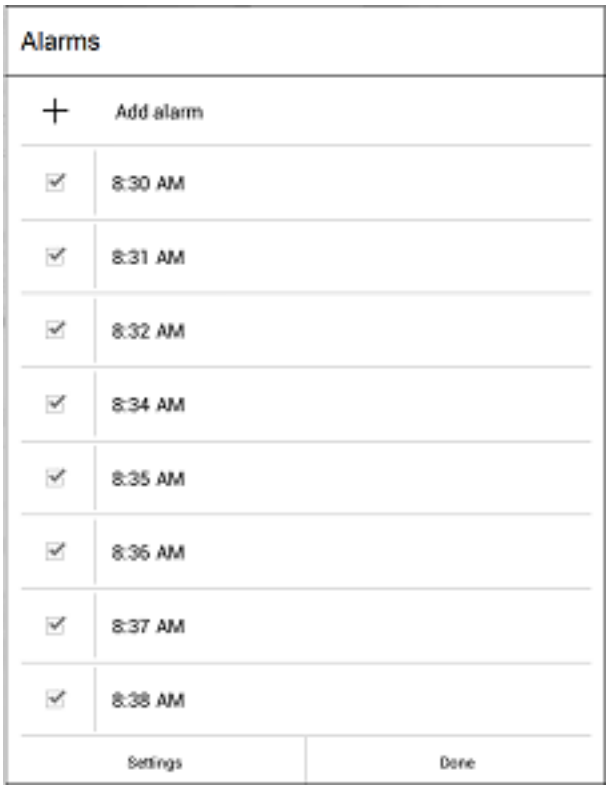
Выходной файл содержит одну строку из 7 целых чисел через пробел. K-ое число характеризует исправность K-ого элемента блока: для исправного элемента оно равно 0, в случае неисправности типа A равно 1, в случае неисправности типа B равно -1.

Примеры

input.txt	output.txt
.xxx.....xxx...xxx.....xxx...xxx...xxx...xxx...xxx. x...x.....x.....x.....x.x...x.x...x.x...x.....x.x...x.x...x x...x.....x.....x.....x.x...x.x...x.x...x.....x.x...x.x...x x...x.....x.....x.....x.x...x.x...x.x...x.....x.x...x.x...x ....xxx...xxx...xxx...xxx...xxx.....xxx...xxx. ....x.....x.....x.....x.....x.....x.....x.....x.....x ....x.....x.....x.....x.....x.....x.....x.....x.....x ....x.....x.....x.....x.....x.....x.....x.....x.....x .xxx.....xxx...xxx.....xxx...xxx.....xxx...xxx.	0 0 1 0 -1 1 0

Задача L. Эпилог

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт



Крис открыл глаза и сразу же прищурился. Окно его спальни выходило на восток, и солнечные лучи пробивали не слишком плотные шторы. Он взглянул на экран телефона — да, звонки ему всё-таки не снились: пять пропущенных с одного неизвестного номера. Спустя секунду телефон зазвонил вновь.

— Привет, Крис! Это Саймон. Моя секретарь Элис не смогла до тебя дозвониться. Мне нужно бы обучить сотрудников моей фирмы. Язык $K\pm$, ты же его знаешь. Я уже им рассказал про тебя, они жаждут знаний! — у Саймона явно было хорошее настроение. — Так что ты мне скажешь на этот раз?

«А не такой уж и странный был сон...» — подумал Крис.

— Знаешь, Саймон, у меня в ближайшее время много работы по одному проекту и несколько командировок. Так что не получится.

— А может, получится? Откажись от командировок, брось свой проект, мой точно прибыльнее!.. Нет? Вот всегда ты так. Позову спецов из «Квадротеха», потом жалеть будешь!..

Крис подумал, что сегодня Саймон сделал доброе дело — разбудил его. За последние дни Крис очень устал и, чтобы не проспать, выставил на своём телефоне N будильников на время T_1, T_2, ..., T_N соответственно.

Крис называет степенью усталости количество будильников, которые он проспит.

Ваша задача — определить момент времени, в который проснётся Крис, если степень его усталости равна U.

Формат входного файла

В первой строке содержатся целые числа N ($1 \le N \le 100$) и U ($1 \le U \le 100$) через пробел.

Во второй строке содержатся целые числа $T_1, T_2, \ldots T_N$ ($0 \le T_1, T_2, \ldots, T_N \le 10000$) — моменты времени, на которые выставлены будильники. Все моменты времени попарно различны.

Формат выходного файла

В первой строке выведите момент времени, в который проснётся Крис. Если он поставил недостаточное количество будильников, выведите -1.

Примеры

input.txt	output.txt
5 3 832 831 835 833 834	834
4 7 1032 944 1112 807	-1