Задача A. Канарейки

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

На днях в Московский зоопарк прибыли новые жильцы — целых N канареек. Пока бедные птенцы томятся в неудобных временных контейнерах, в зале заседаний зоопарка на Совете орнитологов решается их судьба. А именно, ученым предстоит решить, как лучше всего распределить N канареек по имеющимся в зоопарке K клеткам так, чтобы при этом ни одна клетка не пустовала. Поскольку главным критерием при размещении птиц является комфорт, орнитологов в первую очередь интересует, сколько канареек окажется в самой заполненной клетке (то есть в клетке с максимальным числом канареек). Для начала, Вам, как главному (и, как это ни печально, единственному) программисту зоопарка, поручили оценить эту величину, то есть найти, какое минимально и максимально возможное количество птиц может оказаться в самой заполненной клетке при условии, что ни одна клетка не останется пустой.

Формат входного файла

В единственной строке содержатся два натуральных числа, разделенных пробелом: N — количество канареек и K — количество клеток ($1 \leqslant K \leqslant N \leqslant 10^9$).

Формат выходного файла

Выведите два натуральных числа: минимально и максимально возможное количество канареек в самой заполненной клетке.

Примеры

stdin	stdout
7 4	2 4
12 3	4 10

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
- Тесты 3–9. В тестах этой группы $N = K$. Эта группа оценивается в 30 баллов.
- Тесты 10–16. В тестах этой группы N делится на K. Эта группа оценивается в 30 баллов.
- Тесты 17–23. В тестах этой группы дополнительные ограничения отсутствуют. Решение оценивается в 40 баллов.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

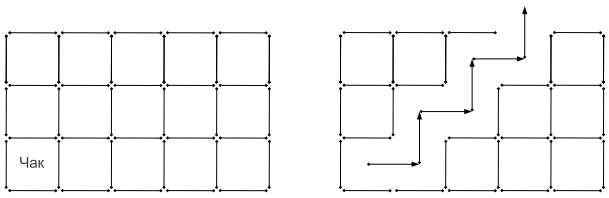
Задача B. Страусиная ферма

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

Как вы помните, Джонни работает в правительственных службах одной неизвестной страны. В свободное от государственных заданий время он разводит страусов на своей маленькой ферме.

На ферме есть $N \times M$ птиц. Джонни соорудил каждому страусу по загону, установив перегородки так, чтобы они образовывали прямоугольник из N строк и M столбцов. Тем самым образуется ровно $N \times M$ квадратных загонов 1×1. Обратите внимание: между соседними загонами он ставил ровно одну перегородку, а не две.

В один прекрасный осенний день заслуженный страус Чак, находившийся в нижнем левом загоне, почувствовал острую необходимость отправиться по важным и неотложным страусиным делам. Он начал пробивать себе путь на волю, ломая перегородки. Сначала он сломал правую перегородку и переместился загоном правее. Потом он сломал верхнюю перегородку и переместился вверх. Далее он прокладывал себе путь по такому же принципу: ломая попеременно то правую, то верхнюю перегородку, пока, наконец, не оказался на свободе.



Джонни, увидев разгром, учиненный Чаком, сильно расстроился. Но делать нечего — надо приводить все в порядок. Он отправил письмо на ближайшую лесопилку, указав, сколько у него осталось перегородок, но забыв при этом указать, сколько ему требуется.

Помогите работникам лесопилки: зная, сколько у Джонни осталось перегородок, определите, каких размеров могла быть ферма.

Формат входного файла

В единственной строке входного файла задано целое число X — количество перегородок, оставшихся у Джонни ($1 \leqslant X \leqslant 10^9$).

Формат выходного файла

В первой строке выходного файла выведите C — число возможных вариантов размеров фермы Джонни.

В последующих C строках выведите возможные варианты размеров фермы. В каждой строке выведите через пробел два целых числа — возможное значение N и M.

Обратите внимание, Джонни мог ошибиться при подсчете оставшихся перегородок, поэтому возможно, что не существует вариантов, подходящих под условие. В таком случае требуется вывести единственное число 0.

Примеры

stdin	stdout
9	2 3 1 2 2

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тест 1. Тест из условия, оценивается в ноль баллов.
- Тесты 2–15. В тестах этой группы $X \leqslant 15$. Решение оценивается в 30 баллов.

- Тесты 16–35. В тестах этой группы $X \leqslant 1\,000$. Решение оценивается в 30 баллов.
- Тесты 36–55. В тестах этой группы дополнительные ограничения отсутствуют. Решение оценивается в 40 баллов.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача C. Кафельная плитка

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

Иннокентий устроил ремонт на кухне. Как профессиональный строитель, он прекрасно знает, что для кухни нет ничего лучше кафельной плитки.

Кухня Иннокентия представляет собой прямоугольник W на H метров. К сожалению, нужная плитка продается только в одном магазине. Каждая плитка имеет фиксированный размер a на b метров, и на нее нанесен интересный узор. Для того, чтобы пол кухни выглядел красиво, плитку надо класть так, чтобы каждая сторона плитки граничила максимум с одной плиткой и была параллельна одной из сторон кухни. Узор является очень специфическим, поэтому плитки **нельзя поворачивать**, даже все одновременно — сторона кухни длиной W должна быть всегда параллельна стороне плитки длиной a.

Возможно, плитки придется разрезать на меньшие части с помощью прямолинейных разрезов вдоль одной из сторон. При этом полученные части плитки можно также разрезать на меньшие части. Иннокентий хочет замостить кухню так, чтобы в итоге было использовано минимально возможное количество плиток и их частей.

Помогите Иннокентию выяснить, какое минимальное число целых плиток размером a на b нужно купить, чтобы красиво замостить всю кухню.

Формат входного файла

В первой строке входных данных содержатся два целых числа W и H — размеры кухни ($1 \leqslant W, H \leqslant 10\,000$). В следующей строке содержится два целых числа a и b — размеры одной плитки ($1 \leqslant a \leqslant W$, $1 \leqslant b \leqslant H$).

Формат выходного файла

Выведите одно число — минимальное число плиток, которое необходимо купить Иннокентию. Помните, что плитки ни в коем случае нельзя поворачивать!

Примеры

stdin	stdout
10 10 2 2	25
3 5 2 2	4
35 17 25 1	26

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–3. Тесты из условия, оцениваются в ноль баллов.
- Тесты 4–6. В тестах этой группы W делится на a и H делится на b. Эта группа оценивается в 30 баллов.
- Тесты 7–9. В тестах этой группы W делится на a. Эта группа оценивается в 30 баллов.
- Тесты 10–20. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача D. Нумерация серий

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

Создатели мультсериала «Футурама» при разработке серий используют нумерацию, которая называется «код серии». Все серии разбиты на производственные блоки. Код серии имеет вид: **xACVyy**, где x — номер производственного блока (возможно, двузначный), а yy — номер серии в блоке (обязательно двузначный, возможно, с ведущим нулем). При показе по телевизору используется другая нумерация, называемая «код показа». Код показа имеет вид **SxxEyy**, где xx — номер сезона, а yy — номер серии в сезоне. Числа xx и yy обязательно двузначные, возможно, с ведущим нулем. Порядок и суммарное количество серий при создании и показе совпадает.

По информации о количестве производственных блоков, сезонов и количестве серий в каждом из блоков и сезонов необходимо создать «словарь», в котором каждому коду серии будет сопоставлен код показа.

Формат входного файла

В первой строке задается два числа N и M ($1 \leqslant N, M \leqslant 50$) — количество производственных блоков и сезонов соответственно.

Во второй строке содержится N чисел, задающих количество серий в каждом из N блоков. Количество серий в каждом блоке не меньше 1 и не больше 50.

В третьей строке содержится M чисел, задающих количество серий в каждом из M сезонов. Количество серий в каждом сезоне не меньше 1 и не больше 50.

Формат выходного файла

Для каждой серии выведите код серии и код показа, разделенные пробелом. Вывод необходимо осуществлять в том порядке, в котором создавались серии.

Примеры

stdin	stdout
2 3 3 2 2 1 2	1ACV01 S01E01 1ACV02 S01E02 1ACV03 S02E01 2ACV01 S03E01 2ACV02 S03E02

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тест 1. Тест из условия, оценивается в ноль баллов.
- Тесты 2–10. В тестах этой группы $N = 1$. Эта группа оценивается в 30 баллов.
- Тесты 11–20. В тестах этой группы $M = 1$. Эта группа оценивается в 30 баллов.
- Тесты 21–30. В тестах этой группы дополнительные ограничения отсутствуют. Решение оценивается в 40 баллов.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача E. Распродажа

Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

В супермаркете «На троечку» часто происходят распродажи товаров, срок годности которых подходит к концу. Каждый товар привозят в магазин в определенное время, а через некоторое его вывозят из магазина, в связи с окончанием срока годности. Более формально, каждый товар имеет стоимость c_i, время его завоза в магазин a_i и время его вывоза из магазина b_i.

У Иннокентия есть хитрый план похода в магазин. Даже несколько. Каждый план похода в магазин выглядит так: Иннокентий выбирает какое-то время, когда он появится в магазине m_j, время s_j, которое он проведет в магазине среди огромных стеллажей товаров, и сумму денег k_j, которую он рассчитывает потратить. Для каждого плана он хочет узнать, сможет ли он осуществить его, т. е. верно ли, что он сможет во время своего пребывания в магазине купить несколько товаров суммарной стоимостью **ровно** k_j, при этом все выбранные товары должны быть в магазине на протяжении всего пребывания Иннокентия в магазине.

Помогите Иннокентию определить, какие из его планов можно выполнить.

Формат входного файла

В первой строке входных данных содержится число N — общее количество товаров в магазине ($1 \leqslant N \leqslant 500$). Далее содержатся описания товаров, каждый товар описывается тремя целыми числами c_i, a_i, b_i, обозначающими стоимость товара, время его завоза и время его вывоза из магазина ($1 \leqslant c_i \leqslant 1\,000$, $1 \leqslant a_i < b_i \leqslant 10^9$).

Далее содержится число M — количество планов Иннокентия ($1 \leqslant M \leqslant 500\,000$). Каждый план описывается тремя целыми числами m_j, k_j, s_j, обозначающими время прихода Иннокентия в магазин, сумму денег, которую он готов потратить в этом плане и длительность его пребывания в магазине ($1 \leqslant m_j \leqslant 10^9$, $1 \leqslant k_j \leqslant 100\,000$, $0 \leqslant s_j \leqslant 10^9$).

Помните, что это только планы, т. е. ситуация в магазине не меняется вне зависимости от того, может ли Иннокентий осуществить план или нет.

Формат выходного файла

Для каждого плана в отдельной строке выведите «YES», если Иннокентий может его осуществить, и «NO» в противном случае.

Примеры

stdin	stdout
5 6 2 7 5 4 9 1 2 4 2 5 8 1 3 9 5 2 7 1 2 7 2 3 2 0 5 7 2 4 1 5	YES NO YES YES NO

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тест 1. Тест из условия, оценивается в ноль баллов.
- Тесты 2–7. В тестах этой группы $M \leqslant 10$, $a_i, b_i, m_j, s_j \leqslant 20\,000$, $k_j \leqslant 1\,000$. Эта группа оценивается в 30 баллов.

- Тесты 8–11. В тестах этой группы $N \leqslant 200$, $M \leqslant 200$, $k_j \leqslant 5\,000$. Эта группа оценивается в 30 баллов.
- Тесты 12–23. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех тестов из второй и третьей групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача F. Числа в Зазеркалье

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

Назовем N-значное число, не содержащее ведущих незначащих нулей, *числом из Зазеркалья*, если это число можно написать на бумаге, изображая цифры так, как их пишут на электронных табло (см. рисунок), а потом поднести к этому изображению зеркало и увидеть в нем то же самое число. При этом все цифры полностью мы должны увидеть именно в зеркале, в неискаженном виде, а число целиком прочитать, как обычно, слева направо. Единственное, что может выглядеть по-другому, — это расстояние между цифрами числа.

Вася выписал на бумаге некоторые цифры одного из N-значных чисел. Позиции этих цифр в числе он также зафиксировал. Помогите ему определить, сколько различных чисел из Зазеркалья он может записать, заполняя всеми допустимыми способами остальные позиции.

Формат входного файла

В первой строке входных данных записано одно натуральное число N ($1 \leqslant N \leqslant 30$). Во второй строке находятся ровно N символов, часть из которых цифры, а часть — символы '*', обозначающие свободные места. Строка заканчивается символом перевода строки.

Формат выходного файла

Выведите количество N-значных чисел из Зазеркалья, которые можно получить, заполняя свободные места цифрами.

Примеры

stdin	stdout
2 *0	3
4 1*7*	0

Примечание

Условие этой задачи нужно понять *буквально*.

А для того чтобы проверить ответ к первому примеру, можно перебрать все варианты на бумаге и подносить к ним зеркало, пока не станет понятно, какие 3 варианта являются подходящими.

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
- Тесты 3–26. В тестах этой группы N не превосходит 4. Эта группа оценивается в 30 баллов.
- Тесты 27–32. В тестах этой группы N не превосходит 8. Эта группа оценивается в 30 баллов.
- Тесты 33–40. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline**, т. е. после окончания тура, причем только в случае прохождения всех тестов из второй и третьей групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача G. Правила дорожного движения

Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

В столице одной небольшой страны очень сложная ситуация. Многокилометровые пробки буквально парализовали движение в городе, и власти на многих улицах ввели одностороннее движение, не анализируя, можно ли будет теперь проехать из любого места в городе в любое другое, не нарушая правила. Транспортная система столицы представляет собой N площадей, соединенных M полосами для движения, в том числе круговыми полосами, проходящими по площади. Каждая полоса предназначена для движения только в одну определенную сторону. При этом на магистралях есть полосы, направленные как в одну, так и в другую сторону. По круговой полосе можно двигаться только внутри площади и только против часовой стрелки.

Власти города на каждой полосе разместили видеокамеру, поэтому если Иннокентий едет по встречной полосе (при ее наличии) или, в случае одностороннего движения, в сторону противоположную предписанной знаками, то после поездки против правил по каждой из полос ему придется заплатить штраф в размере одной тысячи тугриков этой страны.

Иннокентий, который торопится купить кафельную плитку со скидкой, решился доехать до магазина в любом случае, даже если для этого придется нарушать правила. Но он хочет выбрать такой маршрут движения, суммарный штраф на котором минимален.

Иннокентий еще не решил, откуда именно и в какой магазин он собирается ехать, поэтому ему необходимо ответить на несколько вопросов вида «Какой минимальный штраф надо заплатить, чтобы добраться из пункта A в пункт B?». Отвечая на потребности жителей столицы, известная поисковая система *Индекс* разрабатывает соответствующий сервис.

Так как многие из вас рано или поздно будут проходить собеседование на работу в эту фирму, продемонстрируйте, что вы тоже умеете решать эту задачу.

Формат входного файла

В первой строке входных данных содержатся два числа N и M — количество площадей и полос движения в городе соответственно ($1 \leqslant N \leqslant 5000$, $1 \leqslant M \leqslant 10\,000$). Далее содержатся описания полос, по которым движение разрешено. Каждая полоса описывается номерами двух площадей, которые она соединяет. Движение разрешено в направлении от первой из указанных площадей ко второй.

В следующей строке содержится одно число K — количество вопросов у Иннокентия ($1 \leqslant K \leqslant 10\,000$, $N \cdot K \leqslant 2 \cdot 10^7$). В следующих строках описываются вопросы, каждый вопрос описывается номерами двух площадей, между которыми требуется найти самый дешевый путь. Путь необходимо проложить от первой из указанных площадей ко второй.

Формат выходного файла

Для каждого вопроса выведите одно число — искомый минимальный размер штрафа в тысячах тугриков. В случае, если пути между выбранной парой площадей не существует, выведите -1.

Примеры

stdin	stdout
5 5 2 1 2 4 3 2 4 3 5 4 3 5 1 1 5 2 3	0 2 0

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–1. Тест из условия, оценивается в ноль баллов.
- Тесты 2–10. В тестах этой группы N не превосходит 10, M не превосходит 20. Эта группа оценивается в 30 баллов.
- Тесты 11–20. В тестах этой группы N не превосходит 2000, M не превосходит 3000, K равно 1. Эта группа оценивается в 30 баллов.
- Тесты 21–47. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех тестов из второй и третьей групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача H. Петя отдыхает

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

Петя — большой фанат активного отдыха. Поэтому он решил активно отдохнуть от занятий в школе в течение N дней и посетить K праздничных мероприятий, проходящих в M разных городах. Петя начал свой отдых с того, что определил, за сколько дней он может добраться от каждого города до любого другого, а также распечатал и определил расписание всех мероприятий, которые он хотел бы посетить. В первый день Петя находится в первом городе, там же он должен находиться в день N, иначе его выгонят из школы.

Для каждого мероприятия известен день его начала, окончания, а также город, в котором оно проходит. Петя живёт полной жизнью, поэтому он посещает каждое мероприятие только целиком. Также Петя не может находиться на двух разных мероприятиях одновременно, даже если они проходят в одном городе. Однако, если два мероприятия проходят в одном городе, и первое заканчивается в тот же день, когда начинается второе, Петя может посетить оба. Также он может посетить мероприятие, начинающееся в день его приезда в город или заканчивающееся в день его отъезда из города.

Учитель информатики задал Вам задачку на дом: узнать, какое максимальное количество мероприятий может посетить Петя.

Формат входного файла

В первой строке даны три числа — количество дней активного отдыха N ($1 \leqslant N \leqslant 10^6$), количество городов M ($1 \leqslant M \leqslant 250$) и количество мероприятий K ($1 \leqslant K \leqslant 5000$).

В следующих M строках записаны по M чисел. j-ое число в $i+1$-ой строчке — целое положительное (за исключением диагональных элементов) количество дней $A_{i,j}$, необходимое, чтобы добраться от i-го города до j-го. $A_{i,j} \leqslant N$. Диагональные элементы равны 0. Значение 1 в этой матрице означает, что Петя приедет в соответствующий город на следующий день после отъезда.

В следующих K строках записаны мероприятия. В каждой строке записаны три целых положительных числа G_i, S_i, F_i — номер города мероприятия, а также дни начала и окончания мероприятия соответственно. $1 \leqslant G_i \leqslant M$, $1 \leqslant S_i < F_i \leqslant N$.

Формат выходного файла

Выведите единственное число — максимальное количество мероприятий, которые сможет посетить Петя.

Примеры

stdin	stdout
14 5 5 0 2 1 2 1 2 0 1 1 2 2 1 0 1 2 1 1 2 0 2 1 1 2 2 0 1 11 12 1 8 10 4 7 8 5 7 8 4 9 10	3
14 5 7 0 1 4 3 2 4 0 3 2 4 3 2 0 1 2 1 1 2 0 3 3 2 2 3 0 4 10 12 5 7 9 3 6 9 4 5 6 1 10 12 1 6 9 3 5 8	2

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
- Тесты 3–9. В тестах этой группы $K \leqslant 18$. Эта группа оценивается в 15 баллов.
- Тесты 10–18. В тестах этой группы $K \leqslant 400$. Эта группа оценивается в 25 баллов.
- Тесты 19–33. В тестах этой группы дополнительные ограничения отсутствуют. Решение оценивается в 60 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех тестов из второй и третьей групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача I. Расшифровка

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

Недавно на уроке во время контрольной Мария Ивановна перехватила записку Саше от Оли. Мария Ивановна очень хочет знать, что в записке, но, к сожалению, записка зашифрована. Мария Ивановна знает, что её ученики для шифровки заменяют каждую букву исходного сообщения на какую-то другую. Замена происходит таким образом, что одинаковые буквы всегда заменяются одной и той же буквой, а разные — разными.

Мария Ивановна подозревает, что записка — это ответы к контрольному тесту (ведь её длина случайно оказалась равной длине строки с правильными ответами). Однако она знает, что ответы Оли не обязательно полностью правильны. На каждый вопрос возможен один из K вариантов ответа. Естественно, Мария Ивановна знает правильные ответы.

Мария Ивановна решила расшифровать записку таким способом, чтобы максимизировать количество правильных ответов Оли. Однако, она очень занята, поэтому попросила Вас помочь ей в этом пустяковом деле.

Формат входного файла

В первой строке задана длина каждой из строк N ($1 \leqslant N \leqslant 2\,000\,000$) и K — количество возможных ответов на каждый вопрос ($1 \leqslant K \leqslant 52$). Ответы нумеруются в порядке `abcde...xyzABCDE...XYZ`. То есть, при $K = 6$ возможные ответы выглядят как `abcdef`, а при $K = 30$ — `abcde...xyzABCD`.

Во второй строке задана зашифрованная записка — строка, состоящая из строчных и заглавных латинских букв.

В третьей строке заданы правильные ответы — строка той же длины, что и первая, состоящая из строчных и заглавных латинских букв.

Формат выходного файла

В первой строке выведите единственное число — максимально возможное количество правильных ответов у Оли.

Во второй строке выведите расшифровку — строчку длины K, где по порядку для каждой буквы из шифра учеников указано, какому ответу она соответствует.

Если несколько расшифровок дают правильный ответ, выведите любую.

Примеры

stdin	stdout
10 2 aaabbbaaab bbbbabbbbb	7 ba
10 2 aaaaaaabbb bbbbaaabbb	6 ab
9 4 dacbdacbd acbdacbda	9 cdba

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–3. Тесты из условия, оцениваются в ноль баллов.
- Тесты 4–15. В тестах этой группы $K = 2$. Решение оценивается в 15 баллов.
- Тесты 16–40. В тестах этой группы $K \leqslant 9$. Решение оценивается в 15 баллов.

- Тесты 41–75. В тестах этой группы $K \leqslant 26$. Решение оценивается в 30 баллов.
- Тесты 76–115. Дополнительные ограничения отсутствуют. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех предыдущих групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы. Тестирование на очередной группе начинается только после полного прохождения предыдущей.

Задача J. Страусиные страсти

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

Как вы помните, у Джонни приключился досадный инцидент со страусом Чаком. Поймав всех разбежавшихся страусов, фермер задумался о том, как бы не повторить подобного беспорядка в будущем.

Джонни принял решение построить новое жильё для птиц. Он поставил N наблюдательных вышек с часовыми и прожекторами и соединил их заборами с колючей проволкой под напряжением, огородив тем самым дворик в форме многоугольника из N вершин, на котором будут жить страусы. Для простоты наблюдения он хочет соединить некоторые пары несоседних вышек стенами с пулемётчиками, разбив тем самым двор на $N-2$ треугольных части, образующих загоны для страусов. Естественно, стены должны лежать целиком внутри многоугольника, и каждый загон должен представлять собой невырожденный треугольник, т. е. его площадь должна быть ненулевой.

Сейчас фермеру предстоит выбрать, как именно он будет разбивать двор. Помогите ему проанализировать все возможные варианты проведения стен. Джонни интересуют два вопроса — как водится, попроще и посложнее. Во-первых, он хочет знать, для каких пар вышек существует план разбиения, включающий стену между ними. Во-вторых, его интересует, сколько вообще существует планов разбиения двора.

Формат входного файла

В первой строке входного файла идёт число N ($3 \leqslant N \leqslant 300$) — количество вышек.

В последующих N строках даны описания вышек по порядку их обхода. i-я строка содержит два целых числа x_i, y_i, по модулю не превосходящих 10^4 — координаты i-ой вышки.

Гарантируется, что двор представляет собой многоугольник ненулевой площади без самокасаний и самопересечений.

Формат выходного файла

В первой строке выведите K — количество пар вышек, между которым потенциально может быть проведена стена.

Далее выведите K пар номеров вышек по одному в каждой строке. Числа в каждой паре должны быть упорядочены по возрастанию, пары должны идти сначала по возрастанию первых чисел, потом по возрастанию вторых чисел.

Последняя строка должна содержать единственное число — количество способов поделить двор на загоны. Так как оно может быть достаточно большим, выведите остаток от его деления на $2^{30} = 1073741824$.

Примеры

stdin	stdout
5 1 1 3 3 4 0 2 -2 1 0	5 1 3 1 4 2 4 2 5 3 5 5
6 2 2 1 4 0 2 2 0 4 2 3 4	5 1 3 1 4 1 5 2 4 4 6 4

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
- Тесты 3–18. В тестах этой группы N не превосходит 12. Группа оценивается в 20 баллов.
- Тесты 19–26. В тестах этой группы двор представляет строго выпуклый многоугольник. Группа оценивается в 20 баллов.
- Тесты 27–42. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 60 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех тестов из второй и третьей групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача K. Стулья

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

«Принимая во внимание современные тенденции в области высоких технологий, с 2019 года Всероссийскую олимпиаду по информатике было решено проводить на планшетных компьютерах. Это нововведение настолько популяризовало программирование, что в 2020 году в отборочных этапах олимпиады приняло участие беспрецедентно много участников. Как следствие, количество участников заключительного этапа 2020 года также возросло, и в этом году впервые превысит порог в...»

Вот в такой сложной ситуации оказался оргкомитет заключительного этапа Всероссийской олимпиады по информатике 2020 года. Чтобы максимально комфортно разместить всех участников в зале, оргкомитет принял решение рассадить участников за квадратные столы, до четырех участников за каждый стол: ведь школьнику с планшетом много места для работы не нужно. У каждой стороны стола может стоять максимум один стул, иначе участники будут задевать друг друга локтями.

Всю ночь перед пробным туром отряд дежурных расставлял стулья вокруг столов по одним им известному принципу, и, как казалось, успешно справился с задачей, расставив необходимое количество стульев к утру. Но на пробном туре выяснились две новости, хорошая и плохая. Плохая новость — если два стула стоят спиной друг к другу, то дежурные не могут пройти между ними, поэтому в некоторые точки зала дежурные просто не могут попасть, что противоречит правилам проведения олимпиады. Хорошая новость — очень большой процент заявленных участников просто не доехал до места проведения и принимать участие в заключительном этапе не планирует, так что часть стульев можно просто убрать, и, таким образом, правила будут соблюдены.

Так что всю следующую ночь дежурные снова проведут в зале, теперь уже унося какие-то из лишних стульев, освобождая себе проход. Однако, есть опасность, что они увлекутся и вынесут что-нибудь не то, поэтому старший дежурный решил заранее нарисовать схему расстановки мебели, которой нужно добиться.

Схема должна отличаться от текущей расстановки мебели только отсутствием некоторых стульев. Мебель должна быть расставлена так, чтобы дежурные могли пройти в любую пустую точку зала, при этом из зала должно быть убрано как можно меньше стульев.

Формат входного файла

На вход подаются числа N и M — количество столов, помещающихся по ширине и по длине зала, а также план расстановки мебели в прямоугольном зале. Зал заполнен столами полностью, то есть столов всего $N \times M$. План представляет собой таблицу размером $3N \times 3M$, где каждый стол с его окружением задан квадратом 3×3.

В квадрате, задающим стол со стульями вокруг, стол обозначается латинской буквой T, стул — латинской буквой C, пустое место — символом ".". Гарантируется, что стол всегда стоит в центре такого квадрата. Стул может стоять только у одной из четырех сторон стола.

В первой строке через пробел вводятся числа N ($1 \leqslant N \leqslant 100$) и M ($1 \leqslant M \leqslant 100$). В следующих $3N$ строчках задается текущая расстановка мебели согласно указанному выше формату. Каждая строчка имеет длину $3M$.

Формат выходного файла

Выведите план расстановки мебели в зале без лишних стульев в формате, аналогичному формату входных данных.

Примеры

stdin	stdout
2 2 ...... .TCCT. .C..C. .C..C. .TCCT.	...... .T.CT. .C..C. .C..C. .TCCT. ......
1 1 .C. CTC .C.	.C. CTC .C.

Примечание

Тесты к этой задаче состоят из трех групп.

- Тесты 1–2. Тесты из условия, оцениваются в ноль баллов.
- Тесты 3–20. В тестах этой группы $N, M \leqslant 5$. Эта группа оценивается в 30 баллов.
- Тесты 21–50. В тестах этой группы дополнительных ограничений нет. Эта группа оценивается в 70 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех тестов из второй группы.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы.

Задача L. Phigon

Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

В последнее время стал очень популярным язык программирования «Phigon». Главная особенность этого языка состоит в том, что все программы на этом языке выглядят очень коротко и лаконично. К сожалению, эти программы отличаются тем, что иногда очень трудно понять, почему же они работают долго.

Вам поручено выяснить ответ на этот вопрос. Для этого первым делом необходимо написать анализатор программы, который определяет, сколько в ней выполняется действий. Мы будем рассматривать упрощенную модель программы на языке «Phigon». Примеры программ на языке «Phigon» смотрите в конце условия.

Как и в любом языке программирования, в «Phigon» есть переменные. Названия переменных — это строки из маленьких латинских букв длиной не более 50. Переменные могут иметь любые названия, кроме «`if`», «`while`», «`or`», «`and`» и «`not`».

Как и в любом языке программирования, в «Phigon» есть арифметические выражения, состоящие из целых чисел, переменных и знаков арифметических операций «`+`», «`-`» и «`*`». Более формально, арифметические выражения в «Phigon» задаются следующим образом:

```
<арифметическое выражение> ::= <слагаемое> | <слагаемое> (+|-) <арифметическое выражение>
<слагаемое> ::= <множитель> | <множитель> * <слагаемое>
<множитель> ::= -<множитель> | <неотрицательное число> | <переменная> | (<арифметическое выражени
```

Тем самым, как и в любом языке программирования, в «Phigon» унарный минус имеет больший приоритет, чем умножение, а оно в свою очередь имеет больший приоритет, чем сложение. Каждое отрицательное число, встречающееся в программе, считается полученным в результате применения унарного минуса к соответствующему положительному числу. Например, запись отрицательного числа вида `-42` означает применение унарного минуса к неотрицательному числу `42`.

Как и в любом языке программирования, в «Phigon» есть операторы присваивания, которые выглядят следующим образом:

```
<оператор присваивания> ::=
<переменная> = <арифметическое выражение>
```

В результате исполнения подобного оператора переменной, стоящей в левой части, присваивается значение арифметического выражения, стоящего в правой части.

Начиная с момента первого присваивания, переменная считается объявленной, и становится возможным дальнейшее её использование в арифметических выражениях.

Как и в любом языке программирования, в «Phigon» есть логические выражения, состоящие из арифметических выражений, знаков сравнения «`<`», «`<=`», «`>`», «`>=`», «`==`», «`!=`» и логических операторов «`and`», «`or`» и «`not`». Более формально,

```
<логическое выражение> ::= <конъюнкция> | <конъюнкция> or <логическое выражение>
<конъюнкция> ::= <логическое условие> | <логическое условие> and <конъюнкция>
<логическое условие> ::= not <логическое условие> | <арифметическое выражение>
(<|<=|>|>=|==|!=) <арифметическое выражение> | (<логическое выражение>)
```

.

Тем самым, как и в любом языке программирования, в «Phigon» отрицание имеет больший приоритет, чем оператор конъюнкции «И», а он в свою очередь имеет больший приоритет, чем оператор дизъюнкции «ИЛИ».

Как и в любом языке программирования, в «Phigon» есть условный оператор, имеющий следующий вид:

```
<условный оператор> ::=
if <логическое выражение>:
<отступ><блок операторов>
```

где <блок операторов> — это некоторая последовательность операторов, идущая с дополнительным отступом в 4 пробела. Формальное определение для <блок операторов> мы дадим позднее. Как и в любом языке программирования, если логическое выражение истинно, то соответствующий блок операторов исполняется, а если ложно — то нет. Заметим, что при наличии вложенного условного оператора, у соответствующего блока операторов уже будет отступ в 8 пробелов, и так далее.

Как и в любом языке программирования, в «Phigon» есть оператор цикла, имеющий следующий вид:

```
<оператор цикла> ::=
while <логическое выражение>:
<отступ><блок операторов>
```

где, как и в условном операторе, <блок операторов> — это некоторая последовательность операторов, идущая с дополнительным отступом в 4 пробела. Как и в любом языке программирования, блок операторов исполняется снова и снова, пока верно логическое выражение.

Тем самым мы, наконец, можем определить общий вид программы на языке программирования «Phigon». Программа представляет собой блок операторов присваивания, условных операторов и операторов цикла:

```
<блок операторов> ::= <оператор> | <оператор> <блок операторов>
<оператор> ::= <оператор присваивания> | <условный оператор> | <оператор цикла>
```

Ваша задача состоит в том, чтобы узнать значение переменных после выполнения программы и количество раз, которое была выполнена каждая из операций `=`, `+`, `-` (суммарно как бинарный и как унарный), `*`, `<=`, `>=`, `<`, `>`, `==`, `!=`, `and`, `not`, `or`.

Формат входного файла

Во входном файле задана программа на языке «Phigon». Каждая инструкция расположена на своей строке. Пустых строк нет. Длина каждой строки не превосходит 500. В программе не более 5000 инструкций. Гарантируется, что значения всех промежуточных вычислений и численных констант в программе по модулю строго меньше 10^{500}.

Гарантируется, что перед использованием каждой переменной в арифметических выражениях её значение было определено в операторе присваивания.

Формат выходного файла

Выведите информацию о количестве раз, которые вызваны операции в следующем формате: для каждой операции, которая была выполнена, выведите строку `операция:` c, где c — это количество раз, которое была вызвана эта операция. Информацию об операциях надо выводить в лексикографическом порядке (обратите внимание на второй пример).

После этого выведите строку «`total N operations`», где N — суммарное количество выполненных операций. Гарантируется, что общее число операций не превосходит 200 000.

После этого через пустую строку выведите информацию о значении переменных после выполнения программы в формате `Имя переменной:` v, где v — значение переменной в порядке лексикографического возрастания имен переменных.

Примеры

stdin	stdout
a=3 b=a+7*a	*: 1 +: 1 =: 2 total 4 operations a: 3 b: 24

a = 1
if (a > 0):
  b=a+a*a-(2*a+(-a)*5)
if (a>=0)and (a < 2):
  b = b + 1
if (a<=2) and (a ==1) and(a !=666):
  b = b + (-3)*(-3)*0*(-3) + 1
if ((not (a>4)) or (a>4) or (a>4)):
  b = b + 1
	!=: 1
	*: 6
	+: 6
	-: 5
	<: 1
	<=: 1
	=: 5
	==: 1
	>: 4
	>=: 1
	and: 3
	not: 1
	or: 2
	total 37 operations
	
	a: 1
	b: 8

x = 1 while x < 5: x = x + 1	+: 4 <: 5 =: 5 total 14 operations x: 5

Примечание

Тесты к этой задаче состоят из четырех групп.

- Тесты 1–3. Тесты из условия, оцениваются в ноль баллов.
- Тесты 4–26. В тестах этой группы нет инструкций `if` и `while`. Эта группа оценивается в 30 баллов.
- Тесты 27–39. В тестах этой группы нет инструкций `while`. Эта группа оценивается в 30 баллов.
- Тесты 40–49. В тестах этой группы дополнительные ограничения отсутствуют. Эта группа оценивается в 40 баллов. Решение будет тестироваться на тестах этой группы **offline,** т. е. после окончания тура, причем только в случае прохождения всех тестов из второй и третьей групп.

Баллы за каждую группу тестов ставятся только при прохождении **всех** тестов группы и **всех предыдущих групп**.

Задача M. Охота на короля

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Максим и Лёша поехали на сборы в Петрозаводск. После тура они решили не ходить на доре-шивание, а поиграть в шахматы. Игра проходит на бесконечной доске, и в какой-то момент Максим хитрыми махинациями добился того, что у него осталось три белых ладьи, а у Алексея — один чёрный король.

После своего поражения Алексей заподозрил, что Максим в некоторый момент сжульничал (ну откуда на бесконечной шахматной доске взяться третьей белой ладье, да и куда подевался король Максима?). Чтобы восстановаить справедливость он потребовал реванша и предложил разыграть ситуацию, когда у Максима осталось две белых ладьи и король, а у Лёши — чёрный король.

Помогите Максиму при таких условиях поставить мат Лёше и доказать тем самым, что он просто лучше играет в шахматы, а состав фигур ни на что не влияет.

Формат входного файла

Это интерактивная задача. При запуске решения на стандартный поток ввода поступают 8 чисел — координаты чёрного короля, двух белых ладей и белого короля на поле. Координаты не превосходят по модулю 10. Гарантируется, что в начальный момент никакие две фигуры не стоят на одной клетке, чёрный король не находится под боем ни одной из ладей, и короли не атакуют друг друга. Первыми делают ход белые фигуры, то есть начинает Максим.

На каждый ход Максима вводится ответный ход Алексея — перемещение короля d_x, d_y относительно текущей позиции ($0 \leqslant |d_x|, |d_y| \leqslant 1$). В случае, если $|d_x| = |d_y| = 0$, программа должна немедленно завершиться (это означает, что был поставлен мат, пат или сделан некорректный ход).

Формат выходного файла

Для каждого хода выводите на стандартный поток вывода три числа — обозначение фигуры (ладьи обозначаются как `"R1"` и `"R2"`, король — как `"K"`) и перемещение этой фигуры l_x, l_y ($|l_x|+|l_y| > 0$). Ход должен быть корректным, т. е. фигура не должна пойти в занятую клетку, ладья не может перепрыгнуть через другую фигуру, король не может пойти в клетку, атакуемую вражеским королём. *Перемещение ладьи не должно быть больше, чем на* 100 *клеток.*

Примеры

stdin	stdout
3 2 0 3 2 0 1 0 0 -1 -1 1 1 0 -1 0 0 0	R2 2 0 R1 4 0 R2 0 1 K 1 0 R2 0 1

Примечание

Вывод должен завершаться переводом строки и сбросом буфера потока вывода. Для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Программа не должна делать более 80 ходов. Если после хода программы мат не поставлен, а король сделать ход не может, то считается, что тест не пройден.

Ладья может ходить на произвольное количество клеток по горизонтали или по вертикали. Король же может ходить в любую соседнюю клетку, которая не находится под боем ладьи, в том числе, он может «съесть» ладью, если клетка, в которой ладья стоит, не находится под боем другой ладьи. Обратите внимание, что в случае, если король съест ладью, то выигрыш станет невозможным, и тест не будет пройден.

Мат — ситуация в шахматах, когда король находится под ударом ладьи, а игрок не может сделать ни одного хода, чтобы его избежать. Пат — ситуация в шахматах, когда король не находится под ударом ладьи, но при этом игрок не может сделать ни одного хода.

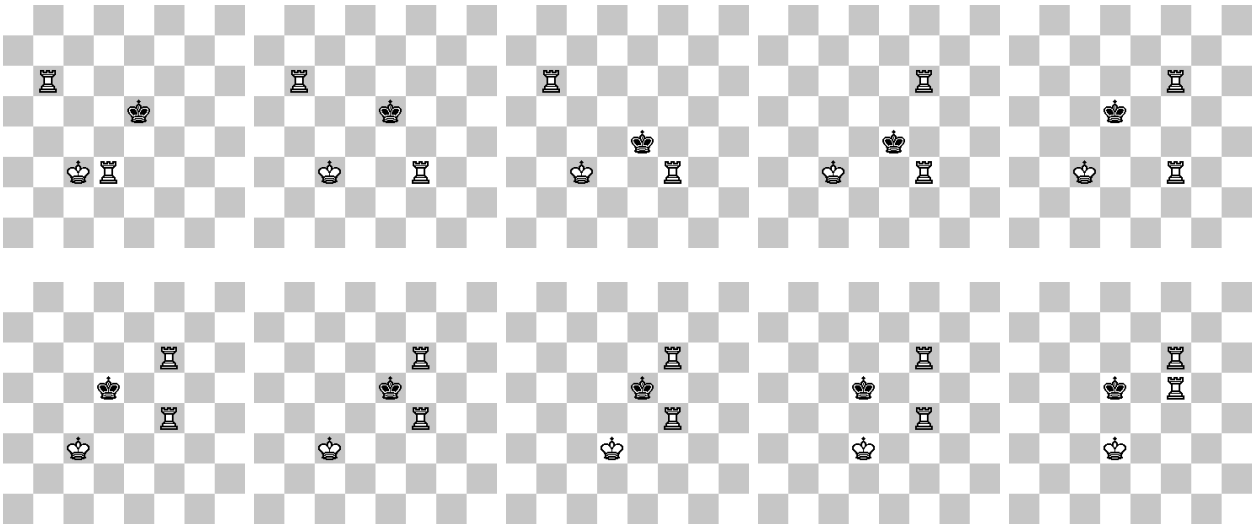
Тесты к этой задаче состоят из нескольких групп партий. Баллы за каждую группу ставятся только при выигрыше всех партий в группе. Группы оцениваются независимо. Вам будут доступны результаты тестирования на первых двух группах: они соответствуют тестам с номерами 1–13 и 14–26. Каждая из этих групп оценивается в 20 баллов.

Оставшиеся 60 баллов соответствуют группам, тестирование на которых будет происходить **Offline**, т. е. после окончания олимпиады.

Обратите внимание, что тестирующая система сообщает подробный протокол взаимодействия вашей программы и программы жюри. Гарантируется, что существует программа, которая на всех тестах жюри умеет ставить мат.

Пример ответа тестирующей системы для примера из условия.

```
Starting position: black king on (3 2), white rooks on (0 3) (2 0), white king on (1 0).

Turn 1
Max moves 2nd rook to (4 0)...
OK.
Alex moved king to (3 1)
Black king on (3 1), white rooks on (0 3) (4 0), white king on (1 0)

Turn 2
Max moves 1st rook to (4 3)...
OK.
Alex moved king to (2 2)
Black king on (2 2), white rooks on (4 3) (4 0), white king on (1 0)

Turn 3
Max moves 2nd rook to (4 1)...
OK.
Alex moved king to (3 2)
Black king on (3 2), white rooks on (4 3) (4 1), white king on (1 0)

Turn 4
Max moves king to (2 0)...
OK.
Alex moved king to (2 2)
Black king on (2 2), white rooks on (4 3) (4 1), white king on (2 0)

Turn 5
Max moves 2nd rook to (4 2)...
OK.
It's nowhere for king to go :-(
Game over! Checkmate - Max wins!
```



Положения фигур по ходу партии из примера.