Задача Departure. Отъезд

Имя входного файла:	`departure.in`
Имя выходного файла:	`departure.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Близится время отъезда и, чтобы он получился организованным, каждый ЛКШонок должен знать номер автобуса на котором он поедет в Москву. В этом году ожидаются настолько вместительные автобусы, что каждый из них способен вместить всех ЛКШат.

Автобусов будет ровно два. Зачем два? Дело в том, что про некоторых ЛКШат мы знаем, что их ни в коем случае нельзя сажать в один автобус. Про других ЛКШат мы наоборот знаем, что они обязательно должны быть в одном автобусе.

Помогите нам распределить ЛКШат по автобусам.

Формат входного файла

В первой строке входного файла находится число n ($1 \leqslant n \leqslant 10\,000$) — количество ЛКШат. Во второй строке находится число m ($1 \leqslant m \leqslant 100\,000$) — количество пар ЛКШат на которые администрация будет обращать особое внимание при распределении по автобусам. Следующие m строк содержат по три целых числа i, j и k каждая ($1 \leqslant i, j \leqslant n; 1 \leqslant k \leqslant 2$). Если k равно одному, то ЛКШата i и j должны обязательно сидеть в одном автобусе. Если k равно двум, то ЛКШата i и j должны обязательно сидеть в разных автобусах.

Формат выходного файла

В первой строке выходного файла выведите количество детей в первом автобусе. Во второй строке через пробел выведите номера ЛКШат, которые будут сидеть в первом автобусе. Если рассадка невозможна, то выведите `-1`. Если существует несколько рассадок, то выведите любую.

Примеры

departure.in	departure.out
2 2 1 2 1 1 2 2	-1

Задача Numinc. Возрастающая подпоследовательность

Имя входного файла:	`numinc.in`
Имя выходного файла:	`numinc.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Задана последовательность целых чисел. Найдите количество ее возрастающих подпоследовательностей.

Формат входного файла

Первая строка входного файла содержит длину последовательности n ($1 \leqslant n \leqslant 100$), а вторая — ее элементы (натуральные числа, меньшие 5000).

Формат выходного файла

Выведите количество возрастающих подпоследовательностей по модулю 10^6.

Пример

numinc.in	numinc.out
3 1 2 3	7
3 3 1 2	4

Задача Firesafe. Противопожарная безопасность

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

В городе Судиславль n домов. Некоторые из них соединены дорогами с односторонним движением.

В последнее время в Судиславле участились случаи пожаров. В связи с этим жители решили построить в городе несколько пожарных станций. Но возникла проблема — едущая по вызову пожарная машина, конечно, может игнорировать направление движения текущей дороги, однако, возвращающаяся с задания машина обязана следовать правилам дорожного движения (жители Судиславля свято чтут эти правила!).

Ясно, что где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой выехала. Но строительство станций стоит больших денег, поэтому на совете города было решено построить минимальное количество станций таким образом, чтобы это условие выполнялось. Кроме того, для экономии было решено строить станции в виде пристроек к уже существующим домам.

Ваша задача — написать программу, рассчитывающую оптимальное положение станций.

Формат входного файла

В первой строке входного файла задано число n ($1 \leqslant n \leqslant 3\,000$). Во второй строке записано количество дорог m ($1 \leqslant m \leqslant 100\,000$). Далее следует описание дорог в формате a_i b_i, означающее, что по i-й дороге разрешается движение автотранспорта от дома a_i к дому b_i ($1 \leqslant a_i, b_i \leqslant n$).

Формат выходного файла

В первой строке выведите минимальное количество пожарных станций K, которые необходимо построить. Во второй строке выведите K чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

Пример

firesafe.in	firesafe.out
5 7 1 2 2 3 3 1 2 1 2 3 3 4 2 5	2 4 5

Задача Signchange. Знакочередование

Имя входного файла:	`signchange.in`
Имя выходного файла:	`signchange.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Реализуйте структуру данных из n элементов $a_1, a_2 \ldots a_n$, поддерживающую следующие операции:

- присвоить элементу a_i значение j;
- найти знакочередующуюся сумму на отрезке от l до r включительно ($a_l - a_{l+1} + a_{l+2} - \ldots \pm a_r$).

Формат входного файла

В первой строке входного файла содержится натуральное число n ($1 \leqslant n \leqslant 10^5$) — длина массива. Во второй строке записаны начальные значения элементов (неотрицательные целые числа, не превосходящие 10^4).

В третьей строке находится натуральное число m ($1 \leqslant m \leqslant 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задается тремя числами `0 i j` ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant 10^4$).
- операция второго типа задается тремя числами `1 l r` ($1 \leqslant l \leqslant r \leqslant n$).

Формат выходного файла

Для каждой операции второго типа выведите на отдельной строке соответствующую знакочередующуюся сумму.

Пример

signchange.in	signchange.out
3 1 2 3 5 1 1 2 1 1 3 1 2 3 0 2 1 1 1 3	-1 2 -1 3

Задача Prefix. Последнее слово Джека

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

Джек недавно прочитал на заборе занимательное и новое для него слово. Оно настолько понравилось Джеку, что он захотел сам придумать ещё какое-нибудь занимательное слово. Но только ничего у него не вышло — все придуманные им слова состояли из префиксов исходного слова и поэтому не приносили радости. Он стал придумывать всё более и более длинные слова, но ни одно из них не было оригинальным...

И вот настало время Джеку сказать своё последнее слово.

Формат входного файла

Первая строка содержит занимательное слово, которое было написано на заборе. Вторая строка содержит последнее слово Джека. Длины слов не превосходят 75 000, слова непустые и состоят из строчных латинских букв.

Формат выходного файла

Если Джек так ничего и не придумал своего, выведите первой строкой `No`. В этом случае покажите Джеку, как разбить его последнее слово на несколько частей, каждая из которых является исходным словом или его непустым префиксом — выведите все эти части во второй строке, разделяя их пробелом. Если же такого разбиения нет, и последнее слово было за Джеком, выведите единственной строкой `Yes`.

Примеры

prefix.in	prefix.out
abracadabra abrabracada	No abr abracad a
abracadabra arbadacarba	Yes

Задача Matrix. Матрица

Имя входного файла:	`matrix.in`
Имя выходного файла:	`matrix.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мебибайта

Вам дана квадратная матрица $n \times n$ из нулей и единиц. Напишите программу, которая находит две одинаковые непересекающиеся подматрицы размера $k \times k$.

Подматрица задается своим левым верхним углом (i, j). В этом случае она состоит из элементов

$$\begin{matrix} a_{i,j} & a_{i,j+1} & \dots & a_{i,j+k-1} \\ a_{i+1,j} & a_{i+1,j+1} & \dots & a_{i,j+k-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i+k-1,j} & a_{i+k-1,j+1} & \dots & a_{i+k-1,j+k-1} \end{matrix}$$

Формат входного файла

В первой строке входного файла находятся числа n и k ($1 \leqslant k \leqslant n \leqslant 1\,000$). В каждой из последующих n строк находится по n чисел. В i-ой строке на j-ом месте находится целое число $a_{i,j}$ ($0 \leqslant a_{i,j} \leqslant 1$).

Формат выходного файла

Если две искомые подматрицы существуют, то в первой строке выведите координаты верхнего левого угла первой подматрицы, а во второй строке — координаты верхнего левого угла второй подматрицы.

Если искомых подматриц не существует, выведите единственное число `-1`.

Пример

matrix.in	matrix.out
4 2 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0 0	3 1 1 2
3 2 0 1 1 1 1 1 1 1 0	-1

Задача Folding. Свертка

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

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть ее повторяющиеся подпоследовательности. Например, последовательность `AAAAAAAAAABABABCCD` может быть записана как `10(A)2(BA)B2(C)D`.

Формальной определение свернутой последовательности и соответствующей ей операции развертки дается следующим образом:

- Последовательность, которая содержит единственный символ от `'A'` до `'Z'` представляет из себя свернутую последовательность. При развертке такой последовательности получается она сама.
- Если S и Q — свернутые последовательности, то SQ также свернутая последовательность. Если при развертке строки S получается строка S', а при развертке Q получается Q', то при развертке SQ получается строка $S'Q'$.
- Если S — свернутая последовательность, то $X(S)$ также свернутая последовательность, где X это десятичное представление целого числа большего единицы. Если при развертке строки S получается строка S', то при развертке $X(S)$ получается строка S', повторенная X раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

Формат входного файла

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 100 символов.

Формат выходного файла

В выходной файл выведите одну строку, содержащую наименьшую последовательность развертка которой даст строку, заданную во входном файле.

Если ответов несколько — выведите любой из них.

Пример

folding.in	folding.out
AAAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

Задача Army. Военный парад

Имя входного файла:	army.in
Имя выходного файла:	army.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

> Главное — не победа, а участие в дележе призовых
>
> Девиз НОК Шотландии

Скив с недоумением перечитывал текст приглашения:

«Приглашаем представителей Поссилтума принять участие в традиционном открытом первенстве Универсума по программированию. Спонсор соревнований — Казначейство — в этом году установило для победителя приз. . . »

— И что мне на это отвечать? — спросил он.

— А, ты про приглашение? — откликнулся Ааз. — Ну как, я думаю, что исходя из суммы приза, приглашение надо принять. Глупо упускать такие деньги.

— А кого же мы туда отправим?

— В качестве основного участника поедешь ты. Вроде бы ты в последнее время не только магию, но и программирование изучал, да и судя по этим текстам — Ааз показал на пачку листов с задачами, на которых сверху было написано «SPb SU Championship 2006–2007» — ты в этих олимпиадах уже своим человеком стал. В помощь возьмёшь Гвидо и Нунцио, тогда команда будет солиднее смотреться. Я поеду с вами как тренер команды.

Проводы делегации в Поссилтуме были организованы на высшем уровне. Генерал Хью Плохсекир даже организовал военный парад. Особенно впечатляли перестроения. Каждое такое перестроение выглядело следующим образом: из шеренги длины n попарно различных по росту солдат, построенных в некотором порядке A, выходило $2k$ подряд стоящих солдат, затем по расчёту на первый-второй вышедшие менялись местами так, что те, кто занимал среди вышедших $2i-1$-е места, стали занимать $2i$-е, и наоборот, здесь $1 \leqslant i \leqslant k$. После этого солдаты уже в новом порядке возвращались на своё место в шеренге.

Скив заинтересовался, за какое минимальное количество подобных перестроений можно из порядка A получить порядок B. Генерал затруднился ответить: в уставе ничего подобного не сказано. А сможете ли ответить на этот вопрос вы?

Формат входного файла

В первой строке задаётся число солдат n ($1 \leqslant n \leqslant 8$). Во второй строке даны n чисел — порядковые номера солдат, образующих построение A, в третьей также n чисел — порядковые номера солдат, образующих построение B.

Формат выходного файла

В первой строке выведите минимальное число перестроений l. Далее выведите l строк, в каждой из которых записана пара a_j k, обозначающая, что надо применить перестроение к группе из $2k$ солдат, начинающейся с позиции a_j в общей шеренге.

Примеры

army.in	army.out
3 1 2 3 3 2 1	3 1 1 2 1 1 1
4 2 1 4 3 1 2 3 4	1 1 2

Задача King. Король (*)

Имя входного файла:	king.in
Имя выходного файла:	king.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мебибайта

В Тридесятом царстве, Тридевятом государстве жил-был король. И было у короля n сыновей. В Тридесятом царстве жили n прекрасных девушек, и король знал, какие девушки нравятся каждому сыну (поскольку сыновья были молодыми и безшабашными, то им могли нравиться несколько девушек одновременно).

Однажды король приказал своему советнику подобрать для каждого сына прекрасную девушку, на которой тот сможет жениться. Советник выполнил приказ и подобрал для каждого сына для женитьбы прекрасную девушку, которая ему нравилась. Разуемеется, каждая девушка может выйти замуж только за одного из сыновей.

Посмотрев на список невест, король сказал: «Мне нравится этот список, но я хочу знать для каждого сына список всех девушек, на которых он может жениться. Разумеется, при этом все сыновья также должны иметь возможность жениться на девушках, которые им нравятся».

Эта задача оказалася для советника слишком сложной. Помогите ему избежать казни, решив ее.

Формат входного файла

Первая строка входного файла содержит число n — количество сыновей ($1 \leqslant n \leqslant 2\,000$). Следующие n строк содержат списки прекрасных девушек, которые нравятся сыновьям. В начале идет k_i — количество девушек, которые нравятся i-му сыну. Затем идут k_i чисел — номера девушек. Сумма k_i не превышает 200 000.

Последняя строка входного файла содержит список, составленный советником — n различных чисел от 1 до n: для каждого сына — номер прекрасной девушки, на которой он может жениться. Гарантируется что список корректен — то есть каждому сыну нравится выбранная для него девушка.

Формат выходного файла

Выходной файл должен содержать n строк. Для каждого сына выведите l_i — количество различных девушек, на которых он может жениться. После этого выведите l_i чисел — номера девушек в произвольном порядке.

Пример

king.in	king.out
4 2 1 2 2 1 2 2 2 3 2 3 4 1 2 3 4	2 1 2 2 1 2 1 3 1 4