

Задача А Доброе мороженое

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

По дороге в школу Петя любит забегать в киоск и покупать себе мороженое.

Город можно представить как N перекрестков, соединенных M улицами, про каждую улицу известна ее длина. Дом Пети находится на перекрестке A , школа - на перекрестке B , а киоск с мороженым - на перекрестке C . По пути в школу Петя не хочет проходить через один перекресток дважды.

Входные данные

Первая строка входного файла содержит числа N и M ($3 \leq N \leq 30\,000$, $0 \leq M \leq 50\,000$). Вторая строка содержит три различных числа - A , B и C . Следующие M строк содержат по два целых числа X_i и Y_i - номера перекрестков, соединенных улицей.

Выходные данные

Если путь из дома Пети до школы, проходящий через перекресток с мороженым и не проходящий по одному перекрестку два раза, существует, выведите «yes», иначе «no».

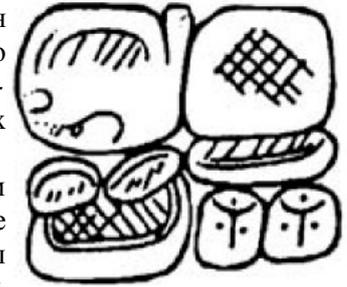
Пример

Входные данные	Выходные данные
5 6 1 5 3 1 2 2 3 2 4 3 4 5 1 4 5	yes

Задача В Добрый иероглиф

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Когда археологи проводили раскопки в древних городах Майя, они обнаружили множество непонятных иероглифов. Пример иероглифа показан справа, он обозначает имя *Кьяк-у-накал*, это имя военного и религиозного лидера в древнем городе Майя *Чичен-Итца* (см. А.В.Восс, Г. Дж.Кремер *Кьяк-у-накал, Хун-пик-токь и Коком*). Этот иероглиф можно увидеть во многих местах древнего города.



Вообще говоря, иероглифы Майя не являются иероглифами в прямом смысле этого слова, а, скорее, являются композицией отдельных глифов. Все известные глифы занумерованы целыми числами от 1 до 9999. Учеными был разработан специальный язык, с помощью которого можно представлять иероглиф в виде обычного текста. Например, иероглиф *Кьяк-у-накал* кодируется как “((669:604).(586:(27:(534.534))))”.

Приведем формальную грамматику этого языка:

<block>	<glyph id>'('<block>'.<horizontal group>)'('<block>'.<vertical group>)'
<horizontal group>	<block>['.<horizontal group>]
<vertical group>	<block>['.<vertical group>]

Код иероглифа описывает процесс его составления. Глифы комбинируются горизонтально и вертикально (с помощью '.' и ')') в блоки, которые, в свою очередь, комбинируются во все большие и большие блоки, до тех пор, пока не будет достигнута нужная конфигурация.

Как обычно, в процессе реализации забыли о важной части — обратном восстановлении обычного текста в иероглиф. Это предстоит сделать вам.

Входные данные

Единственная строка входного файла содержит строку, описывающую иероглиф Майя в виде обычного текста. Длина строки не превышает 255 символов. Строка не содержит пробелов.

Выходные данные

Выведите текст, составленный из символов '+', '-', '|', ' ' (ASCII коды 43, 45, 124, 32), '0'..'9' и переводов строки.

Все блоки одной группы должны иметь одинаковый размер. Номер глифа (glyph id) с одним пробелом перед ним, должен быть помещен в левый верхний угол блока. Вывод должен быть как можно короче. Гарантируется, что для всех тестов существует изображение, содержащее максимум 100 000 байт.

Пример

Входные данные	Выходные данные
((669:604).(586:(27:(534.534))))	+-----+-----+ 669 586 +-----+-----+ 604 27 +-----+ 534 534 +-----+-----+

Задача С Злой лифт

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Дизайн-студия Артемия Индюкова получила заказ на разработку очень пафосного лифта для нового небоскреба. За работу взялся сам Артемий, отличающийся, кстати, редкой неадекватностью. У него есть идея-фикс: для управления лифтом достаточно четырех кнопок. Кнопки должны быть следующие:

Подняться на А этажей вверх

Подняться на В этажей вверх

Подняться на С этажей вверх

Спустится на первый этаж

Изначально лифт находится на первом этаже. Пассажир лифта использует первые три кнопки чтобы попасть на тот этаж, на который он хочет. Если пассажир пытается подняться вверх на А, В или С этажей, а такого этажа в здании не существует (т.е. пассажир хочет подняться выше N-го, последнего этажа), то лифт никуда не едет.

Заказчики проекта оказались с юмором и вместе с отказом от футуристичного дизайна решили оценить адекватность Артемия по шкале от 1 до N. Оценка адекватности равна количеству этажей, на которые можно попасть с первого с помощью такого лифта. Помогите им в этом.

Входные данные

Первая строка содержит число N – высоту небоскреба ($1 \leq N \leq 10^{18}$).

Вторая строка содержит три числа А, В и С, задающие параметры кнопок ($1 \leq А, В, С \leq 100\,000$).

Выходные данные

Выведите единственное число — оценку адекватности Артемия Индюкова.

Примеры

Входные данные	Выходные данные
15 4 7 9	9

Задача D Снова язык Велулу

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Мистер Форд Транкинкс, известный археолог, недавно обнаружил в самом сердце Африки остатки загадочного поселения. Через несколько недель исследования он и его коллеги поняли, что они нашли нечто необычное. Племена Велулу, которые жили там 20 000 лет назад, по-видимому, достигли высокого уровня развития. У них даже был алфавит! Но большинство из них стало жертвой ледникового периода и все их культурные достижения были утрачены. Только несколько счастливиц из племени выжили и попытались возродить исчезнувшую цивилизацию. Теперь они известны под именем грозных зусулов, но об этом в другой раз...

Нынешняя задача состоит в расшифровке текстов Велулу. Основная проблема заключается в том, что в их алфавите нет пробелов. Поэтому все тексты слипаются в одну последовательность букв, которые сложно разобрать.

К счастью, археологи уже разработали проект словаря велульского языка. Конечно же они знают о том, что можно обработать последовательность букв на компьютере и получить слова, имея в своем распоряжении словарь. Однако, воспользовавшись таким методом они обнаружили, что огромное количество таких последовательностей почти для любого текста разумного размера. Они не знают, является ли это проблемой метода обработки или особенностью велульского языка. Поэтому они придумали другой метод, который базируется не только на словаре, но и на порядке частей речи в предложении.

Теперь у них был не только словарь, но и предположение о том, как предложение строится в велульском языке. Вам предстоит определить, сколькими способами можно разбить текст на слова, следуя этим правилам, и вывести один из примеров разбиения.

Входные данные

Вам будет дан словарь, правила построения предложений и текст для расшифровки. Для каждого слова определено, какой частью речи оно может быть.

Первая строка входного файла содержит числа N , M и K , где $1 \leq N \leq 5000$ — количество слов в велульском словаре, $1 \leq M \leq 10$ — количество правил построения предложения, а $1 \leq K \leq 10$ — количество различных частей речи.

В велульском языке не так много букв. Так что археологи закодировали их маленькими латинскими буквами. В каждой из N следующих строк содержится слово (не длиннее 20 символов), затем число k_i ($1 \leq k_i \leq 10$) — количество частей речи, которыми может быть это слово, а затем k_i чисел a_{ij} обозначающие допустимые части речи. Все числа a_{ij} даны в возрастающем порядке. Слова в словаре даны в произвольном порядке (не успели упорядочить). Каждое слово встречается в словаре только один раз.

Следующие M строк содержат правила конструирования предложений. Каждое правило описывается количеством слов в предложении l_i ($1 \leq l_i \leq 10$) и, затем, перечислены l_i чисел, являющихся идентификаторами частей речи. Правила не повторяются.

Последняя строка ввода содержит текст, который надо расшифровать. Текст не пуст и его длина не превышает 1000 символов.

Выходные данные

Первая строка должна содержать количество вариантов расшифровки. Если их больше чем 10^{18} выведите строку «TOO MANY» вместо количества.

Если существует корректное разбиение текста на слова, то выведите одно из них в следующей строке. Исходный текст требуется разделить пробелами и точкой с пробелом для получения набора корректных предложений, каждое из которых удовлетворяет правилам, а все слова содержатся в словаре. Точка должна стоять сразу за окончанием предложения, а все слова должны быть разделены пробелами. Весь текст должен быть разбит на предложения. Внимательно посмотрите пример.

Если вариантов несколько — выведите любой из них.

Пример

Входные данные	Выходные данные
5 2 2 ba 1 2 za 2 1 2 a 2 1 2 caba 1 1 ab 1 1 2 1 2 3 2 2 1 abazabacaba	2 ab a. za ba caba.

Задача Е Охлаждение реактора

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Известная террористическая группа под руководством знаменитого террориста Бен Гадена решила построить атомный реактор для получения оружейного плутония. Вам, как компьютерному гению этой группы, поручили разработать систему охлаждения реактора.

Система охлаждения реактора представляет собой набор труб, соединяющих узлы. По трубам течет жидкость, причем для каждой трубы строго определено направление, в котором она должна по ней течь. Узлы системы охлаждения занумерованы от 1 до N . Система охлаждения должна быть спроектирована таким образом, чтобы для каждого узла за единицу времени количество жидкости, вытекающей в узел, было равно количеству жидкости, вытекающей из узла. То есть если из i -го узла в j -ый течет f_{ij} единиц жидкости за единицу времени (если из i в j нет трубы, то положим $f_{ij} = 0$), то для каждого узла i должно выполняться

$$\sum_{j=1}^N f_{ij} = \sum_{j=1}^N f_{ji}$$

У каждой трубы имеется пропускная способность c_{ij} . Кроме того, для обеспечения достаточного охлаждения требуется, чтобы по трубе протекало не менее l_{ij} единиц жидкости за единицу времени. То есть для трубы, ведущей из i -го узла в j -ый должно выполняться $l_{ij} \leq f_{ij} \leq c_{ij}$.

Вам дано описание системы охлаждения, выясните, каким образом можно пустить жидкость по трубам, чтобы выполнялись все указанные условия.

Входные данные

Первая строка входного файла содержит числа N и M – количество узлов и труб ($1 \leq N \leq 200$). Следующие M строк содержат описание труб. Каждая строка содержит четыре целых числа i, j, l_{ij} и c_{ij} . Любые два узла соединены не более чем одной трубой, если есть труба из i в j , то нет трубы из j в i , никакой узел не соединен трубой сам с собой, $0 \leq l_{ij} \leq c_{ij} \leq 10^5$.

Выходные данные

Если решение существует, выведите на первой строке выходного файла слово YES. Затем выведите M чисел – количество жидкости, которое должно течь по трубам, числа должны быть выведены в том порядке, в котором трубы заданы во входном файле. Если решения не существует, выведите NO.

Пример

Входные данные	Выходные данные
4 6 1 2 1 2 2 3 1 2 3 4 1 2 4 1 1 2 1 3 1 2 4 2 1 2	NO
4 6 1 2 1 3 2 3 1 3 3 4 1 3 4 1 1 3 1 3 1 3 4 2 1 3	YES 1 2 3 2 1 1

Задача F Количество делителей

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

В этой задаче вам требуется найти минимальное положительное целое число, у которого ровно k делителей. Если оно превышает 10^{18} , искать его не нужно.

Входные данные

Во входном файле записано единственное число k ($1 \leq k \leq 50000$).

Выходные данные

В случае, если минимальное число ровно с k делителями больше 10^{18} , выведите -1. В противном случае выведите его.

Пример

Входные данные	Выходные данные
1	1
2	2
6	12

Задача G Функция Аккермана

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Функция Аккермана широко известна всем специалистам по теории вычислений. Это функция двух положительных целых чисел, определённая следующим образом:

$$A(1, j) = 2^j$$

$$A(i, 1) = A(i - 1, 2)$$

$$A(i, j) = A(i - 1, A(i, j - 1))$$

Это не обычная экспоненциальная функция, так как она растёт быстрее, чем любая из них.

Ваша задача – по заданному значению t вычислить $A(n, m) \bmod t$ для нескольких пар n и m . Здесь символом $x \bmod y$ обозначен остаток от целочисленного деления x на y — такое целое r ($0 \leq r < y$), для которого существует целое q , что $x = qy + r$.

Входные данные

Первая строка входного файла содержит целое число t ($1 \leq t \leq 100$). Далее следует несколько строк, каждая из которых содержит пару из n и m ($1 \leq m, n \leq 100$). Последняя строка входного файла содержит два нуля, её обрабатывать не нужно.

Выходные данные

Для каждой пары чисел n и m выведите в выходной файл строку, содержащую её номер и число $A(n, m) \bmod t$. При выводе используйте формат, соответствующий примеру.

Пример

Входные данные	Выходные данные
3	Case 1: 1
3 3	Case 2: 2
2 7	Case 3: 0
1 18	
0 0	

Задача Н Мосты

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Одна сказочная страна располагалась в дельте далекой реки (far-away river). В стране было n островов и на каждом острове находился город. Города были соединены дорогами. Причем существовал в точности один путь от каждого города до любого другого, возможно проходящий через другие города. К сожалению мосты в этой стране были неизвестны, поэтому для пересечения реки использовались понтоны, поэтому путешествия были некомфортными, т.к. приходилось ездить только на лошадях.

Когда было открыто мостостроительство король решил вместо нескольких понтонов построить мосты, по которым могли бы ездить даже кареты. В силу бедности страны только k мостов могут быть построены.

Вам необходимо выбрать какие k понтонов надо заменить на мосты так, чтобы суммарное время путешествия между всеми парами городов оказалось минимальным. Вы можете считать, что по обычным дорогам можно ехать только на лошади, а по дороге с мостом — в карете, запряженной и несколькими лошадьми.

Входные данные

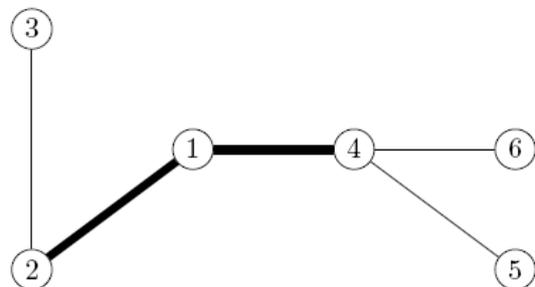
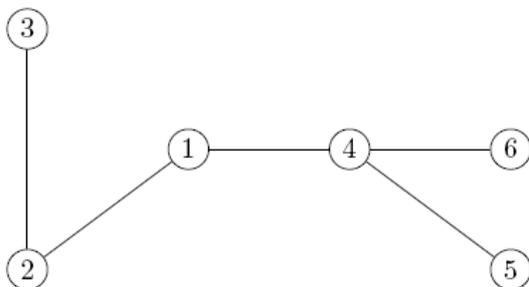
Первая строка входных данных содержит 4 числа n, k, sh и sc — число городов, число мостов, которым можно построить, скорость лошади и скорость экипажа в метрах в секунду ($1 \leq k < n \leq 10\,000, 1 \leq sh; sc \leq 100\,000$). Каждая из следующих $n - 1$ строк содержит три целых числа bi, ei — номера соединяемых городов и длину дороги в метрах li ($1 \leq li \leq 10^6$). Города пронумерованы от 1 до n , дороги пронумерованы от 1 до $n - 1$.

Выходные данные

Выведите k чисел — номера мостов, которые должны быть построены. Если существует несколько оптимальных планов строительства мостов, то выведите любой из них.

Пример

Входные данные	Выходные данные
6 2 1 2 1 2 5 3 2 6 1 4 4 4 6 4 4 5 5	1 3



Задача I Внутренние узлы

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Рассмотрим бесконечную клетчатую бумагу. Покрасим некоторые узлы сетки в черный цвет, а остальные будем считать белыми. Узел V называется *внутренним*, если он *внутренний по вертикали* и *внутренний по горизонтали*. Узел внутренний по горизонтали, если слева и справа от V расположены по крайней мере по одному черному узлу. Узел внутренний по вертикали, если сверху и снизу от V расположены по крайней мере по одному черному узлу.

На каждом шаге все внутренние белые узлы становятся черными, а остальные сохраняют свой цвет. Процесс прекращается, когда все внутренние вершины становятся черными. Напишите программу, которая вычисляет количество черных узлов после окончания процедуры перекраски.

Входные данные

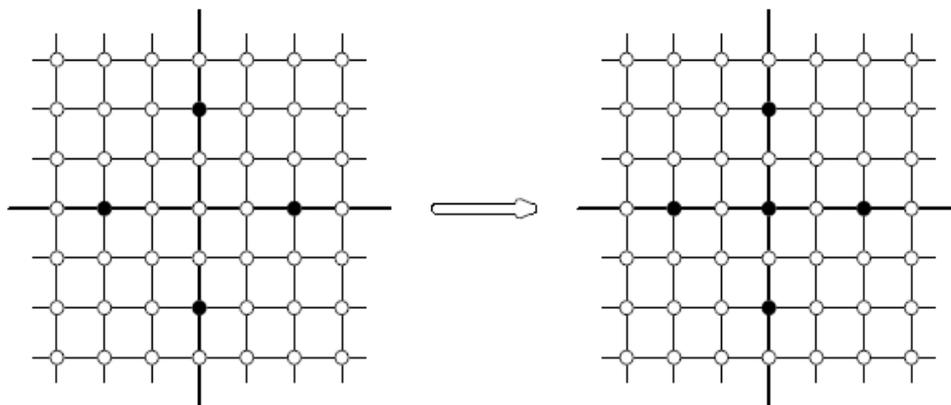
Первая строка содержит одно целое число n ($0 \leq n \leq 100\,000$) – количество черных узлов в начале. Каждая из следующих n строк содержит два целых числа – координаты очередного черного узла, по модулю не превосходящие 10^9 .

Выходные данные

Выведите число черных вершин после окончания всех перекрасок. Если процедура никогда не закончится, то выведите -1 .

Пример

Входные данные	Выходные данные
4 0 2 2 0 -2 0 0 2	5



Задача J Экзамены

Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Первая сессия обычно доставляет много проблем. Одна из них заключается в том, что студенту нужен по крайней мере целый день, чтобы подготовиться к одному экзамену. В день одного экзамена к другому готовиться невозможно. Но основная проблема заключается в том, что студенты не могут начать готовиться к i -му экзамену, не раньше чем за t_i дней до него, иначе они все забудут.

Глеб хочет начать готовиться к экзаменам как можно позже, но он собирается все экзамены сдать. Помогите Глебу выбрать день начала подготовки к экзаменам.

Входные данные

Первая строка выходных данных содержит число экзаменов n ($1 \leq n \leq 50\,000$). Следующие строки описывают экзамены. Каждое описание состоит из трех строк. Первая строка – это название экзамена (строка, содержащая только латинские буквы, длиной не более 10). Вторая строка – дата экзамена в формате dd.mm.yyyy. Третья строка содержит величину t_i для этого экзамена ($1 \leq t_i \leq 100\,000$).

Все экзамены проходят от 01.01.1900 до 31.12.2100. Не забудьте, что високосными считаются годы, которые делятся на 4 и не делятся на 100 или которые делятся на 400.

Выходные данные

Выведите в формате dd.mm.yyyy, когда Глеб самое позднее сможет приступить к подготовке к экзаменам. Если расписание не позволяет подготовиться к каждому из экзаменов, то выведите слово Impossible.

Примеры

Входные данные	Выходные данные
3 Philosophy 01.01.1900 1 Algebra 02.01.1900 3 Physics 04.01.1900 10	30.12.1899
2 Philosophy 29.06.2005 1 Algebra 30.06.2005 2	Impossible