

Задача А. Автобусы

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

В одном городе недавно запустили автобусную сеть. Однако, плата за проезд для жителей этого города оказалась чрезмерной. И несознательные граждане, вместо того, чтобы покупать билет, стали договариваться с водителем и ездить за полцены. Конечно, городская казна понесла серьезные убытки, и было решено взять на работу нескольких контролёров. По уставу, каждый контролёр должен стоять на одном месте и останавливать подозрительные автобусы — с целью проверки билетов.

Для повышения эффективности труда контролёров начальство хочет, чтобы через каждую точку, в которой находится контролёр, проходили маршруты всех автобусов. С другой стороны, нельзя ставить нескольких контролёров в одной точке, чтобы они не отвлекались от выполнения своих обязанностей. Наконец, третья сторона, независимый профсоюз, требует от городской администрации принять на работу максимальное количество контролёров.

Для простоты предположим, что действие происходит на координатной плоскости. Каждый автобус ездит по границе прямоугольника с ненулевыми сторонами, вершины которого имеют целочисленные координаты, а стороны параллельны осям координат. Требуется выяснить, какое максимальное число контролёров удастся принять на работу, если городское управление милиции, в свою очередь, требует, чтобы каждый контролёр находился в точке с целочисленными координатами.

Формат входных данных

На первой строке входного файла находится число n ($1 \leq n \leq 10^4$) — количество маршрутов. Далее следуют n строк, на каждой из которых находятся две пары целых чисел — координаты двух противоположных вершин прямоугольника, по которому проходит данный маршрут. Все координаты не превосходят 10^8 по абсолютной величине.

Формат выходных данных

Выведите в выходной файл одно число — максимальное количество контролёров, которые смогут обрести работу благодаря этому мероприятию.

Пример

<code>buses.in</code>	<code>buses.out</code>
2 0 0 2 2 1 0 4 2	4

Задача В. Часы

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

В марсианских сутках N часов. У марсиан Ятеп и Ашам есть часы со стрелками, которые работают почти так же, как земные — большая стрелка делает один оборот в час, а маленькая — один оборот в сутки. Ятеп и Ашам поссорились и решили не разговаривать, пока стрелки часов не совпадут. Определите точный момент времени, когда это случится.

Формат входных данных

Во входном файле задано число тестов K ($0 \leq K < 10^4$), далее для каждого теста указаны целые числа N , A , B и C ($1 < N < 10^9$, $0 \leq A < N$, $0 \leq B < C < 10^9$). Числа A , B и C означают, что Ятеп и Ашам поссорились в $A + B/C$ часов.

Формат выходных данных

Для каждого теста выведите искомое время в том же формате: числа A , B и C , такие, что искомое время равно $A + B/C$ ($0 \leq A < N$, $0 \leq B < C$, дробь B/C — несократимая).

Пример

<code>clock.in</code>	<code>clock.out</code>
2	0 0 1
12 11 0 1	1 1 11
12 0 0 1	

Задача С. Конкатенация

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

Рассмотрим две строки α и β . Их конкатенацией называется строка, получающаяся в результате приписывания к строке α строки β . Эта строка обозначается $\alpha\beta$. Например, конкатенацией строк $'ab'$ и $'ac'$ будет строка $'abac'$. Очевидно, что это определение естественным образом распространяется на конкатенацию произвольного количества строк. Так, конкатенацией нуля строк будет пустая строка, а конкатенацией одной строки будет она сама.

Рассмотрим некоторое множество W , состоящее из строк. Назовём его замыканием множества W^* , состоящее из тех и только тех строк, которые можно получить в результате конкатенации нуля и более строк из множества W . Таким образом, множество W^* содержит пустую строку, и если строка α принадлежит множеству W^* , а строка β принадлежит множеству W , то строка $\alpha\beta$ принадлежит множеству W^* . Более того, все элементы множества W^* можно представить в таком виде, то есть W^* является пересечением всех множеств с указанными выше свойствами. Например, если $W = \{a, ab\}$, то W^* состоит из всех строк, в которых перед каждой буквой $'b'$ идёт хотя бы одна буква $'a'$.

Задано некоторое множество строк W . Требуется найти множество X , такое, что $W^* = X^*$ и множество X имеет минимальное возможное число элементов. В случае, если таких множеств несколько, подходит любое из них. Например, если $W = \{a, aabb, ab, ac, b, bac\}$, то единственным множеством, удовлетворяющим условиям задачи будет множество $\{a, ac, b\}$.

Формат входных данных

Входной файл состоит из набора строк, каждая из которых является элементом множества W . Каждая строка из множества W встречается во входном файле хотя бы один раз. Суммарная длина всех строк во входном файле не превосходит 10^4 . Количество строк во входном файле не превосходит 10^4 . После каждой строки из множества W во входном файле идёт перевод строки (пара символов с ASCII кодами 13 и 10). Строки состоят из символов с ASCII кодами от 33 до 126 включительно.

Формат выходных данных

Выведите в выходной файл элементы одного из множеств X , удовлетворяющих условиям задачи. Каждая строка множества X должна быть выведена ровно один раз. Строки должны идти в лексикографическом порядке (лексикографический порядок используется в словарях, в этом порядке строка $'ab'$ меньше строки $'aba'$ и строка $'ab'$ меньше строки $'ac'$). После каждой строки множества X должен идти один перевод строки.

Пример

<code>concat.in</code>	<code>concat.out</code>
a	a
aabb	ac
ab	b
ac	
b	
bac	

Задача D. Таможня

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

Идёт 2163 год. Мишу, который работает в отделении таможни при космодроме города Нью-Питер, вызвал в кабинет шеф.

Как оказалось, недавно Министерство Налогов и Сборов выделило отделению определённую сумму денег на установку новых аппаратов для автоматического досмотра грузов. Естественно, средства были выделены с таким расчётом, чтобы грузы теперь находились на таможне ровно столько времени, сколько требуется непосредственно на их досмотр.

В руках шефа каким-то образом оказались сведения о надвигающейся ревизии — список из N грузов, которые будут контролироваться Министерством. Для каждого груза известны время его прибытия, отсчитываемое с некоторого момента, хранимого в большом секрете, и время, требуемое аппарату для обработки этого груза. Шеф дал Мише задание по этим данным определить, какое минимальное количество аппаратов необходимо заказать на заводе, чтобы все грузы Министерства начинали досматриваться сразу после прибытия. Необходимо учесть, что конструкция тех аппаратов, которые было решено установить, не позволяет обрабатывать два груза одновременно на одном аппарате. Напишите программу, которая поможет Мише справиться с его задачей.

Формат входных данных

На первой строке входного файла задано число N ($0 \leq N \leq 50\,000$). На следующих N строках находится по 2 целых положительных числа T_i и L_i — время прибытия соответствующего груза и время, требуемое для его обработки ($1 \leq T_i \leq 10^6$, $1 \leq L_i \leq 10^6$).

Формат выходных данных

В выходной файл выведите одно число — наименьшее количество аппаратов, которое нужно установить, чтобы не вызвать подозрений у Министерства.

Пример

<code>customs.in</code>	<code>customs.out</code>
3	2
3 2	
4 2	
5 2	
5	3
13 4	
15 1	
11 5	
12 3	
10 3	

Задача Е. Динокод

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

Учёные любят присваивать идентификаторы всему живому. Поэтому они обозначают динозавров I эпохи кодом 'a'. Динозавры II эпохи, как произошедшие от динозавров I эпохи, именуется кодом 'aba'. Ящеры III эпохи — 'abacaba', и вообще если $C(n)$ — код динозавров эпохи n , то $C(n+1) = C(n) + S(n+1) + C(n)$, где $S(n+1)$ — символ очередной ($n+1$ -ой) эпохи. Символ первой эпохи — 'a', символ второй эпохи — 'b', затем 'c', 'd', ..., 'x', 'y', 'z'. После букв учёные почему-то перешли на цифры, и обозначили эпохи с XXVII по XXXVI соответственно '0', '1', ..., '9'. После XXXVI эпохи динозавры вымерли, и уже утверждённое название XXXVII эпохи ('a') отдали астрономам для нового кратера на Марсе.

Астрономы (в знак благодарности) нашли какую-то отдалённую звезду с огромной статуей динозавра, похожего на земные аналоги. Экспедиция, посетившая указанную звезду, нашла под статуей надпись, очевидно, с кодом этого динозавра. Впрочем, часть надписи стёрлась. Теперь учёные хотят максимально зависить древность находки. Для этого нужно определить, в коде динозавров какой эпохи — самой древней из подходящих — встречается данный образец (как подстрока). Такую задачу не по силам решить даже астрономам.

Формат входных данных

На первой и единственной строке входного файла находится непустая строка, состоящая из символов 'a', ..., 'z', '0', ..., '9'. Длина строки не превосходит 100.

Формат выходных данных

Выведите два числа — номер эпохи и смещение образца от начала кода. Если же статуя изображает неземного динозавра (или код инопланетян отличается от земного), выведите в выходной файл число 0.

Пример

<code>dinocode.in</code>	<code>dinocode.out</code>
abacabadabacaba	4 0
aca	3 2
e2e4	0

Задача F. Дракон Лун-Ян

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

Петя и его друг Андрейка только что познакомились с китайской мифологией. Особенно им понравились драконы. Поэтому мальчики решили сделать своих драконов из проволоки. Андрейка взял белую проволоку и согнул из неё дракона Лун-Инь: этот дракон спал, свернувшись клубком на столе. Тогда Петя взял чёрную проволоку и согнул дракона Лун-Ян. Этот дракон ничем не походил на Андрейкиного Лун-Иня. Его тело состояло из отрезков прямых, а когда он спал, то сворачивался в виде плоской замкнутой несамопересекающейся ломаной. Более того, Лун-Ян не ложился плашмя на стол для сна, а вставал перпендикулярно поверхности. Удержать равновесие дракон может только тогда, когда существуют две его различные точки, касающиеся стола, такие что центр масс дракона находится строго между ними.

Вам требуется узнать, сколько было устойчивых положений у дракона, в которых он мог сохранять равновесие во время сна, если известно, что форма ломаной в виде которой дракон спит всегда одна и та же.

Формат входных данных

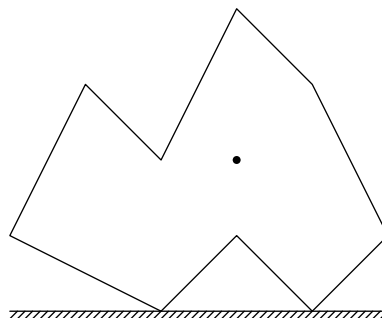
В первой строке входного файла содержится число n ($3 \leq n \leq 1000$) — количество вершин ломаной и два целых числа x_c и y_c — координаты центра масс дракона ($-1000 \leq x_c, y_c \leq 1000$). В следующих n строках содержится по два целых числа x_i и y_i ($-1000 \leq x_i, y_i \leq 1000$) — координаты вершин ломаной в порядке обхода против часовой стрелки (ось OX направлена вправо, а ось OY — вверх).

Формат выходных данных

В первой строке выходного файла выведите число устойчивых положений дракона.

Пример

dragon.in	dragon.out
9 2 2 1 0 2 1 3 0 4 1 3 3 2 4 1 2 0 3 -1 1	4



Задача G. Увлекательная игра

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

Петя и Маша играют в увлекательную игру. Маша загадывает число от 1 до n , записывает его на чистый тетрадный лист, кладёт в конверт и запечатывает. После этого Петя пытается это число отгадать. Он может задавать любые вопросы про это число: “Верно ли, что это число равно трем?”, “Верно ли, что это число — число Фибоначчи?”, “Верно ли, что это число простое?” и так далее. Получив ответ “Да”, Петя отдаёт Маше a конфет, а в случае ответа “Нет” — b конфет.

В какой-то момент Петя произносит сакраментальную фразу: “Я знаю, что это за число”. После этого они распечатывают конверт в присутствии свидетелей, убеждаются в Петевой правоте, и, таким образом, Маша получает внушительную порцию конфет, а Петя — моральное удовлетворение.

Петя очень любит играть в эту игру, но его кондитерские запасы ограничены. Поэтому Петя хочет выяснить, какое минимальное количество конфет может ему потребоваться, чтобы отгадать Машино число в худшем случае. Помогите Пете найти указанный минимум.

Формат входных данных

Входной файл содержит три целых числа: n ($1 \leq n \leq 1000$), a и b ($0 \leq a, b \leq 10^6$).

Формат выходных данных

Выведите одно число — минимальное количество конфет, которое должен иметь Петя, чтобы отгадать Машино число в худшем случае.

Пример

<code>game.in</code>	<code>game.out</code>
8 1 1	3
10 5 0	5
7 0 2	2

Задача Н. Единицы

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

Найдите квадрат числа, десятичная запись которого состоит из n единиц.

Формат входных данных

Входной файл содержит единственное число n ($1 \leq n \leq 10^5$).

Формат выходных данных

Выведите искомый квадрат.

Пример

<code>ones.in</code>	<code>ones.out</code>
2	121
9	12345678987654321

Задача I. Железные дороги

Имя входного файла:	<code>railroad.in</code>
Имя выходного файла:	<code>railroad.out</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

В некоторой стране есть развитая сеть железных дорог. С доисторических времён и до нашего времени в стране непрерывно происходят военные перевороты, из-за которых в системе железнодорожного транспорта этой страны происходят непрерывные изменения. Дело в том, что во время очередного переворота некоторые дороги разрушаются из-за военных действий, а пока новый правитель некоторое время находится у власти, он восстанавливает часть дорог.

Временами железнодорожная система в этой стране становилась довольно разветвленной, поэтому некоторые города могли быть соединены двумя и более дорогами. Кроме того, дорога могла начинаться и заканчиваться в одном и том же городе, причем для одного города таких дорог могло быть несколько.

Инженер Джио проводит испытания новых сверхскоростных поездов. Поскольку поезда экспериментальные, у них не должно возникать трудностей при проезде через промежуточные города. Поэтому инженер Джио требует, чтобы ни в каком городе на пути поезда, кроме, может быть, начального и конечного, не было развилок. Точнее, из любого промежуточного города на пути поезда должны выходить либо ровно две дороги, ведущие в другие города (возможно, в один и тот же), либо ровно одна дорога, начинающаяся и заканчивающаяся в этом городе.

Естественно, что Джио желает испытать поезд на максимальной возможной скорости, и поэтому после каждого изменения в системе путей он хочет знать максимальную длину пути, по которому может ехать поезд. Поскольку в доисторические времена не умели добывать железо, в начале никаких дорог между городами нет.

Формат входных данных

В первой строке входного файла находятся целые положительные числа n ($1 \leq n \leq 500$) — число городов в стране, и m ($1 \leq m \leq 50\,000$) — число изменений в железнодорожной системе. В следующих m строках находится информация об изменениях состояния системы путей. Каждое изменение является либо добавлением дороги, либо удалением дороги. В случае добавления дороги в очередной строке записан ноль, а затем идут три целых числа. Первые два из них являются номерами городов, соединяемых дорогой, а последнее является длиной добавленной дороги. Города нумеруются целыми числами от 1 до n . Длина дороги является целым положительным числом, не превосходящим 10^6 . В случае удаления дороги в очередной строке сначала записана единица, а затем идёт номер шага, на котором произошло добавление удаляемой дороги. Шаги нумеруются целыми числами, начиная с 1.

Формат выходных данных

Для каждого изменения системы путей выведите в очередную строку выходного файла символ '*', если после очередного изменения системы путей существует сколь угодно длинный путь, удовлетворяющий условиям, поставленным Джио. В противном случае выведите в выходной файл единственное целое число, являющееся длиной максимального возможного пути.

Пример

railroad.in	railroad.out
5 16	4
0 2 3 4	7
0 3 4 3	8
0 1 2 1	*
0 5 5 4	*
1 1	3
1 4	8
0 4 1 4	5
0 4 5 1	4
0 1 4 1	3
1 7	8
0 1 5 7	9
1 2	*
1 3	8
1 8	7
1 9	0
1 11	

