Задача A. Мостостроение

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

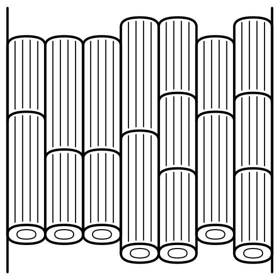
В деревне Гадюкино регулярно идут проливные дожди, в результате чего речка Вонючка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из бревен, оставшихся от строительства бани бизнесмена, поселившегося неподалеку.

Все оставшиеся бревна имеют одинаковую толщину. При этом есть x бревен длины a и y бревен длины b.

Построенный мост должен состоять из l рядов, каждый из которых составлен из одного или нескольких бревен. Пилить бревна нельзя, так как последняя пила утонула при разливе Вонючки.

Главный инженер хочет построить мост максимальной возможной ширины, при этом ширина моста определяется по минимальной ширине ряда бревен.

Например, если нужно построить мост из семи рядов, и при этом есть шесть бревен длины 3 и десять бревен длины 2, то можно построить мост ширины 5.



Формат входного файла

Входной файл содержит пять натуральных чисел: x, a, y, b и l. Все числа не превышают 150. Общее количество бревен не меньше l.

Формат выходного файла

Выведите в выходной файл одно число — максимальную возможную ширину моста.

Примеры

bridge.in	bridge.out
6 3 10 2 7	5
10 7 20 9 25	9

Задача B. Кафе

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

Летнее кафе имеет форму прямоугольника размером $n \times m$ клеток, в каждой из которых может находиться стол или стул.

В кафе планируется проведение банкета на k персон. Для этого требуется расставить ровно k стульев так, чтобы рядом с каждым стулом (в одной из соседних клеток по вертикали, горизонтали или диагонали) был стол. Помогите администраторам кафе решить эту задачу или сообщите, что сделать это невозможно.

Формат входного файла

Входной файл содержит три числа n, m и k ($1 \le n, m \le 50$, $1 \le k \le 10000$).

Формат выходного файла

Выходной файл должен содержать n строк по m символов — план размещения столов и стульев для банкета. Символ «`h`» обозначает стул, символ «`T`» — стол, символ «`.`» — пустую клетку. Если решения нет, выведите строку «`Impossible`».

Примеры

cafe.in	cafe.out

`3 4 8`	`hhh.`
	`hTTh`
	`.hhh`

`2 2 4`	`Impossible`

Задача C. Кот

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

Антон завел себе кота. Кот очень любит спать. Если он засыпает, он спит непрерывно не менее a часов. Более того, долго не спать кот просто не может. Кот не может бодрствовать больше b часов подряд.

Кот мог бы спать все время, но иногда происходят интересные события, пропустить которые кот не может. Например, из школы приходит хозяин, или кота кормят.

Помогите коту спланировать свои сутки так, чтобы не пропустить ни одного интересного события. Каждые сутки кот хочет жить по одному и тому же расписанию.

Формат входного файла

Первая строка входного файла содержит два целых числа a и b ($1 \le a, b \le 24$). Вторая строка входного файла содержит число n — количество интересных событий ($1 \le n \le 20$). Следующие n строк содержат описание интересных событий. Каждое событие описывается строкой вида «`hh:mm-hh:mm`», которая задает период времени, в течение которого оно происходит. Время изменяется от `00:00` до `23:59`. Никакие два интересных события не пересекаются. Если событие завершается раньше, чем началось, это означает, что оно захватывает полночь.

Событие считается занимающим целиком минуту, когда оно начинается и минуту, когда оно заканчивается (например, событие «`12:00-13:59`» продолжается ровно 120 минут). Время начала и время конца события различны.

Формат выходного файла

Если кот может организовать свои сутки так, чтобы во время всех интересных событий не спать, выведите в выходной файл «`Yes`». На второй строке выведите k — сколько раз в сутки кот должен ложиться спать. На следующих k строках выведите интервалы, в которые кот спит в том же формате, в котором интересные события заданы во входном файле. Если решений несколько, выведите любое.

Если кот не может организовать свои сутки искомым образом, выведите в выходной файл «`No`».

Примеры

cat.in	cat.out
3 4 2 07:00-07:15 19:00-19:59	Yes 2 08:00-18:59 20:00-06:59
3 4 3 07:00-08:00 11:00-11:09 19:00-19:59	No
12 12 1 23:00-01:00	Yes 1 01:07-22:13

Задача D. Сборка компьютеров

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

В рамках национальной программы «Электронная Россия» одно государственное учреждение заказало несколько системных блоков и столько же мониторов. При составлении заказа, однако, никто не учел, что существует два типа интерфейсов для соединения системных блоков и мониторов: VGA и DVI. При этом существуют системные блоки и мониторы, которые поддерживают как только один из этих интерфейсов, так и оба.

Поставщик техники оказался не очень добросовестным — в поставке оказалось a_1 системных блоков, которые поддерживают только VGA, a_2 системных блоков, которые поддерживают только DVI, и a_3 системных блоков, которые поддерживают оба интерфейса. С мониторами ситуация аналогична: b_1 мониторов поддерживают только VGA, b_2 — только DVI, b_3 — поддерживают оба интерфейса.

Необходимо выяснить, сколько комплектов из монитора и системного блока можно собрать. При этом соединить монитор и системный блок можно только если у них есть общий интерфейс.

Формат входного файла

Первая строка входного файла содержит три числа a_1, a_2 и a_3 ($0 \le a_1, a_2, a_3 \le 100$). Вторая строка входного файла содержит три числа b_1, b_2 и b_3 ($0 \le b_1, b_2, b_3 \le 100$). При этом выполняется равенство $a_1 + a_2 + a_3 = b_1 + b_2 + b_3$.

Формат выходного файла

В выходной файл выведите максимальное число комплектов из монитора и системного блока, которые можно собрать.

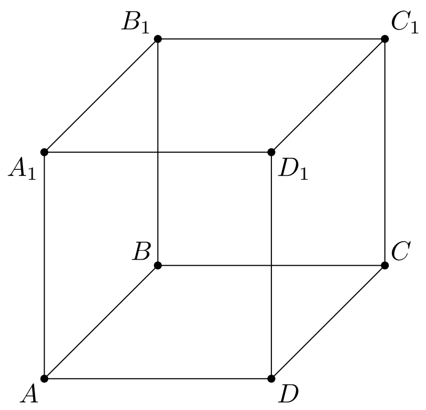
Примеры

computers.in	computers.out
3 4 6 2 3 8	13
3 4 6 2 11 0	12

Задача E. Куб

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

Сегодня на уроке геометрии Маша и Паша изучали геометрические тела. Одним из самых простых геометрических тел является куб. Обозначим четыре вершины нижней грани куба в порядке обхода как A, B, C и D, а четыре вершины верхней грани куба, находящиеся над ними, как A_1, B_1, C_1 и D_1, соответственно.



Машу с Пашей заинтересовал вопрос: при каких условиях отрезки, концы которых находятся в вершинах куба, пересекаются. Помогите им это выяснить.

Формат входного файла

Входной файл содержит две строки. Первая строка содержит названия вершин, которые являются концами первого отрезка, а вторая — названия вершин, которые являются концами второго отрезка. Названия не разделены пробелом, вершины A_1, B_1, C_1 и D_1 задаются как «`A1`», «`B1`», «`C1`» и «`D1`» соответственно.

Отрезки, заданные во входном файле не совпадают. Начало и конец у каждого отрезка не совпадают.

Формат выходного файла

Выведите в выходной файл «`Yes`», если отрезки пересекаются, или «`No`», если нет.

Примеры

cube.in	cube.out
`AA1` `A1C`	`Yes`
`A1C` `DB1`	`Yes`
`AD` `A1C1`	`No`

Задача F. Домашнее задание

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

Сережа очень не любит делать домашние задания, но на последнем уроке информатики учитель задал классу n разных домашних заданий. Причем некоторые домашние задания можно делать только после того, как сделаны некоторые другие.

Для каждого задания Сережа оценил, сколько минут потребуется для его выполнения. После этого Сережа понял, что сделать все задания он точно не успеет. Тогда он решил сделать все задания кроме одного — за одно несделанное задание учитель, наверное, не будет слишком сильно ругаться. Теперь Сереже надо выбрать, какое задание не сделать.

Помогите Сереже выбрать задание, которое можно не делать, чтобы закончить все остальные задания как можно быстрее.

Формат входного файла

Первая строка входного файла содержит целые числа n и m — количество заданий и количество зависимостей между заданиями ($1 \le n \le 100$, $0 \le m \le 1000$). Вторая строка содержит n целых чисел: $t_1, t_2, \ldots, t_n$. Число t_i означает количество минут, необходимое Сереже для выполнения i-го задания ($1 \le t_i \le 1000$).

Затем следует m строк, каждая строка содержит два целых числа. Числа a и b означают, что задание a должно быть выполнено до задания b. Гарантируется, что все задания можно выполнить.

Формат выходного файла

Выведите в выходной файл одно число — минимальное количество минут, необходимое Сереже для выполнения всех заданий кроме одного.

Примеры

homework.in	homework.out
5 5 1 2 3 4 5 1 2 5 3 1 3 3 4 2 4	11

В приведенном примере Сережа может не выполнять четвертое задание. Остальные задания суммарно требуют 11 минут.

Задача G. Москва 2042

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

К 2042 году правительство Москвы завершило очередной масштабный проект, доведя количество кольцевых автодорог до n. Теперь у автомобилиста еще больше способов постоять в пробке в попытке добраться от одной точки города до другой.

Компания «Giggle» планирует в своем новом продукте «Giggle Maps» реализовать возможность проложить оптимальный маршрут от одного перекрестка до другого. Карта Москвы во внутреннем формате программы представляет собой набор из m радиальных и n кольцевых магистралей, при этом некоторые из кольцевых магистралей являются односторонними.

В математической модели «Giggle» все кольцевые магистрали представляют собой концентрические окружности с центром на Красной площади и радиусами $r_1, r_2, \ldots, r_n$. Радиальные магистрали представляют собой отрезки, один из концов каждого отрезка лежит на Красной площади, а другой — на кольцевой магистрали с максимальным радиусом. Если встать на Красной площади и смотреть на восток, то, чтобы посмотреть в направлении j-ой радиальной магистрали, нужно повернуться против часовой стрелки на a_j градусов. По каждой из радиальных магистралей можно ехать в любом направлении. Кольцевые магистрали, в свою очередь, бывают как двусторонними, так и односторонними.

Помогите компании «Giggle» найти кратчайший путь от перекрестка где пересекаются i_s-я кольцевая и j_s-я радиальная магистраль до перекрестка, где пересекаются i_t-я кольцевая и j_t-я радиальная магистраль. При этом проезжать через Красную площадь не разрешается.

Формат входного файла

Первая строка входного файла содержит целые числа n и m ($1 \le n, m \le 100\,000$).

Следующие n строк описывают кольцевые магистрали. Каждая магистраль описывается целым числом r_i ($1 \le r_i \le 10^6$) и числом 0, если магистраль является двусторонней, 1, если по ней разрешено движение только против часовой стрелки (в сторону увеличения углов радиальных магистралей) или -1, если по ней разрешено движение только по часовой стрелке.

Следующие m строк описывают радиальные магистрали. Каждая магистраль описывается одним целым числом A_j, причем $a_j = A_j / 10^6$ ($0 \le A_j < 360 \cdot 10^6$).

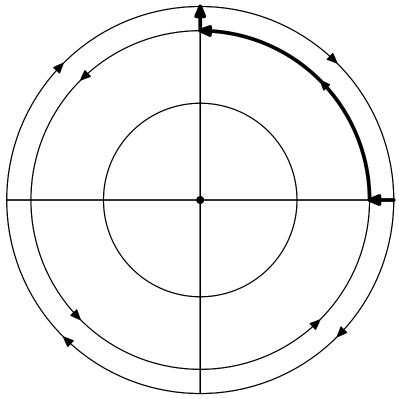
Затем следует две строки: первая из них содержит числа i_s и j_s, а вторая — числа i_t и j_t.

Формат выходного файла

Выведите в выходной файл одно вещественное число: минимальное расстояние, которое придется проехать, чтобы попасть с перекрестка где пересекаются i_s-я кольцевая и j_s-я радиальная магистраль на перекресток, где пересекаются i_t-я кольцевая и j_t-я радиальная магистраль. Ваш ответ должен отличаться от правильного не больше чем на 10^{-4}.

Пример

moscow.in	moscow.out
3 4 1 0 7 1 8 -1 0 90000000 180000000 270000000 3 1 3 2	12.99557428756427634

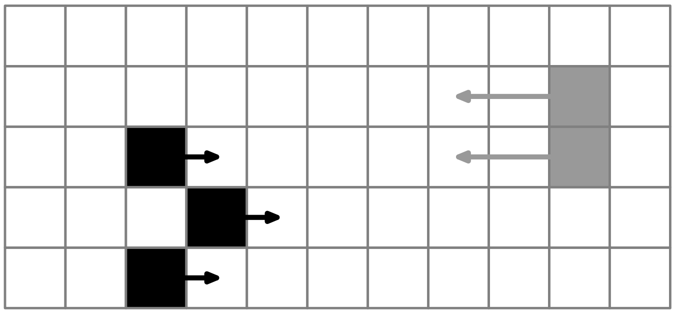


Задача H. Морской бой

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

Недавно Вова купил настольную игру «Морской бой». Это пошаговая игра для двух игроков, действие которой разворачивается на просторах бесконечного клетчатого поля. У каждого из игроков есть свой флот, который состоит из нескольких кораблей. Каждый корабль занимает ровно одну клетку поля. Флот каждого из игроков движется по полю с постоянной скоростью — все корабли i-ого игрока каждые t_i шагов перемещается на вектор $(\Delta x_i, \Delta y_i)$. Таким образом, корабли i-ого игрока перемещаются по полю на шагах с номерами 1, $t_i + 1$, $2 \cdot t_i + 1$ и.т.д.

Если на некотором шаге два корабля оказываются в одной клетке — происходит бой. Бой кораблей является наиболее интересной частью игры, поэтому Вову всегда интересует, через сколько шагов будет первый бой.



Требуется написать программу, которая по заданному расположению кораблей до первого шага игры и их скоростям вычисляет номер шага, на котором будет ближайший бой.

Формат входного файла

Входной файл содержит описания флотов двух игроков. Описание флота состоит из нескольких строк. Первая строка описания содержит четыре целых числа: m_i $(1 \le m_i \le 10000)$ — число кораблей во флоте i-ого игрока, а так же t_i $(1 \le t_i \le 10)$, Δx_i, Δy_i $(|\Delta x_i|, |\Delta y_i| \le 10)$.

Затем следуют m_i строк, каждая из которых содержит по два целых числа — x_j и y_j $(|x_j|, |y_j| \le 10^9)$ — координаты кораблей во флоте до первого шага игры.

Никакие два корабля, описанные во входном файле, не находятся в начальный момент времени в одной и той же клетке.

Формат выходного файла

В выходной файл необходимо вывести номер шага, на котором произойдет первый бой. Если бой никогда не состоится, выведите в выходной файл `-1`.

Примеры

sea.in	sea.out
3 1 1 0 2 0 1 1 1 -1 2 2 -2 0 8 1 8 2	3
1 1 2 0 1 1 1 1 -2 0 2 2	-1

Задача I. Станция «Сортировочная»

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

На железнодорожной станции «Сортировочная» на пути находятся n товарных вагонов, из которых необходимо сформировать состав. Все вагоны имеют одинаковую длину, однако в них размещены различные грузы, поэтому вагоны могут иметь разную массу. Работники железнодорожной станции «Сортировочная» должны выстроить вагоны в порядке возрастания масс — тогда составу будет разрешено отправиться в путь.

Обычно для таких целей используются так называемые маневровые тепловозы и электровозы, однако на этой станции ведутся испытания экспериментального устройства для сортировки вагонов. Предполагается, что оно позволит существенно сократить затраты времени на формирование составов.

Это устройство на воздушной подушке перемещается над вагонами, его длина немного превышает длину двух вагонов. Оно может зависнуть над двумя соседними вагонами, поднять их оба в воздух и поменять местами. Однако, грузоподъемность устройства ограничена — указанную операцию оно может выполнить только, если суммарная масса двух вагонов не превышает M.

Ваша задача состоит в том, чтобы написать программу, которая определит, возможно ли с помощью экспериментального устройства для сортировки вагонов расставить вагоны, находящиеся на пути, в необходимом порядке.

Формат входного файла

Первая строка входного файла содержит два числа: число вагонов n ($2 \le n \le 100\,000$) и грузоподъемность экспериментального устройства M ($1 \le M \le 10^9$). Вторая строка входного файла содержит массы вагонов $m_1, \ldots, m_n$ (для этих масс выполняются неравенства $1 \le m_i \le 10^9$, кроме этого массы вагонов попарно различны). Массы вагонов перечислены во входном файле в том порядке, в котором вагоны исходно стоят на пути.

Формат выходного файла

В выходной файл выведите слово «`Yes`», если с помощью экспериментального устройства для сортировки вагонов можно расставить вагоны в необходимом порядке, и слово «`No`» — иначе.

Примеры

sorting.in	sorting.out
4 10 5 6 3 4	Yes
4 9 5 6 3 4	No

Задача J. Википедия

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

Воодушевленный успехом «Википедии», Петя решил создать аналогичную энциклопедию на своей домашней странице. Поскольку Петя изучает английский язык, он решил сделать английскую версию энциклопедии.

Для начала он взял несколько текстов из «Encyclopedia Britannica» и набрал их. Теперь он хочет расставить внутри статей ссылки на другие статьи. Однако статей очень много, поэтому он решил автоматизировать процесс.

Ссылка на статью на вики-странице устроена следующим образом:

«`[[Название статьи|текст ссылки]]`».

Например, во фразе «`In the wild cats are often enemies of [[Dog|dogs]].`» слово «dogs» будет ссылкой на статью «Dog». Если название статьи совпадает с текстом ссылки, можно ссылку можно оформить просто как

«`[[Название статьи]]`».

Например, во фразе «`Growing together a [[dog]] and a cat can often be friends.`» слово «dog» будет ссылкой на статью «dog». При этом в названиях статей регистр первой буквы игнорируется, а регистр остальных букв важен. Например, слово «dog» может быть ссылкой на статью «Dog», а слово «DOG» — нет.

Помогите Пете расставить ссылки на его сайте. Сайт представляет собой множество статей. Каждая статья имеет название — одно слово, и текст. Для слова названия известны все его словоформы и синонимы.

Будем называть словом в тексте последовательность букв английского алфавита, ограниченную с обеих сторон символами, не являющимися буквами, либо началом или концом строки. В тексте статьи требуется найти все слова, которые являются словоформами или синонимами названий других статей и превратить их в вики-ссылки.

Формат входного файла

Первая строка входного файла содержит число n — количество статей в Петиной википедии ($2 \le n \le 50$). Далее следуют описания статей.

Описание каждой статьи начинается со строки, которая содержит название этой статьи. Далее следует строка, содержащая одно число k — количество словоформ и синонимов к названию статьи, это число не превышает 10. Следующие k строк содержат по одному слову на строке — словоформы и синонимы к названию текущей статьи. Далее следует строка, содержащая число l — количество строк в тексте статьи, это число не превышает 10. Затем следует текст статьи — l строк, каждая из которых имеет длину не более 80 символов.

Все названия статей различны. Все словоформы и синонимы всех названий различны и отличаются от названий статей.

Все слова состоят из букв латинского алфавита, длина каждого слова во входном файле не превышает 20, во входном файле встречаются только пробелы, переводы строк и символы с ASCII кодами от 32 до 126.

Формат выходного файла

Выведите в выходной файл версии статей с расставленными ссылками. Выводите статьи следующим образом. Сначала выведите название статьи. Затем выведите текст статьи, разбитый на строки также как и во входном файле. Все слова в тексте, которые совпадают с названием статьи, или со словоформой или синонимом названия статьи, отличной от той, в которой они встречаются, следует превратить в ссылки. При сравнении слов следует игнорировать регистр первой буквы, но соблюдать регистр остальных. Слова, совпадающие с названием, следует превратить в краткую версию ссылки, а не совпадающие — в полную.

Примеры

wiki.in
2 Cat 2 Kitten Cats 3 Cat is one of the most common pets, together with dogs. In the wild cats are often enemies of dogs. Growing together a dog and a cat can often be friends. Dog 1 Dogs 2 Dog is one of the most common pets, together with cats. DOG can also be an abbreviation.
wiki.out
Cat Cat is one of the most common pets, together with [[Dog|dogs]]. In the wild cats are often enemies of [[Dog|dogs]]. Growing together a [[dog]] and a cat can often be friends. Dog Dog is one of the most common pets, together with [[Cat|cats]]. DOG can also be an abbreviation.