Задача A. Рекламное объявление

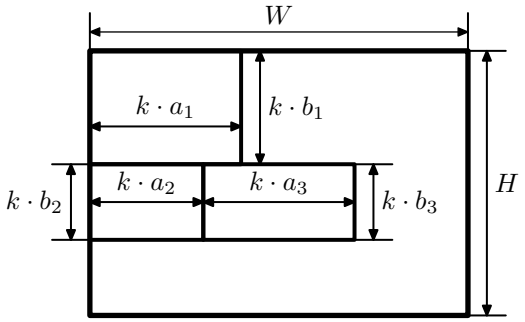
Имя входного файла:	`advert.in`
Имя выходного файла:	`advert.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ивану с детства нравились газеты. У него даже была мечта стать главным редактором газеты. Однажды ему представился шанс осуществить свою мечту. Чтобы устроиться на работу в издательство, ему необходимо выполнить тестовое задание — сверстать рекламное объявление.

Задано поле шириной W и высотой H. Объявление должно состоять из одной или нескольких строк, в которых необходимо разместить в заданном порядке N слов. Про i-е слово известно, что при печати в стандартном масштабе оно занимает прямоугольник шириной a_i и высотой b_i.

Чтобы объявление выглядело красиво, все слова в нем должны быть напечатаны в одном масштабе. При печати в масштабе k размеры всех слов умножаются на k. Если исходно слово занимало прямоугольник $a_i \times b_i$, то при печати в масштабе k оно занимает прямоугольник размером $(k \cdot a_i) \times (k \cdot b_i)$. Кроме того, если в строке более одного слова, то все слова в ней должны иметь одинаковую высоту. Разумеется, ни одно слово не должно выходить за границы поля.

На рисунке приведен пример красивого объявления с тремя словами.



Помогите Ивану найти максимальный масштаб, при котором можно сверстать объявление, которое удовлетворяет этим критериям. Обратите внимание, что менять порядок слов нельзя, они должны читаться по строкам сверху вниз, слева направо в том порядке, в котором заданы.

Формат входного файла

В первой строке входного файла дано три числа: N, W и H ($1 \le N \le 100\,000$, $1 \le W, H \le 10^9$) — число слов в объявлении, длина и высота объявления. В следующих N строках дано по два целых числа, в i-й из них заданы a_i и b_i ($1 \le a_i, b_i \le 10^9$) — ширина и высота i-го слова.

Формат выходного файла

Выведите одно вещественное число k — максимальный масштаб. Ответ требуется вывести с абсолютной или относительной погрешностью не более 10^{-9}. Это значит, что если правильный ответ a, а вы вывели p, то ваш ответ будет засчитан как правильный, если $\frac{|a-p|}{\max(|a|,1)} \le 10^{-6}$.

Примеры

advert.in	advert.out
3 10 7 4 3 3 2 4 2	1.4
2 10 1 2 1 3 2	0.333333333

Задача B. Реклама на заборе

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

Иван живет в небольшом симпатичном домике в деревне. Вдоль его участка расположен забор, который недавно был выкрашен в красный цвет. Но тут в деревню к Ивану пришла цивилизация в лице рекламного агента, расклеивающего всюду свои объявления. И его забор постигла та же участь.

Каждый день на его забор приклеивают новое объявление. Таким образом за последние n дней на забор наклеено уже n объявлений и Ивану кажется, что рекламой заклеен уже весь забор, состоящий из m досок. Доски пронумерованы вдоль забора от 1 до m.

Оказалось, что в каждый из n дней когда приходил рекламный агент и приклеивал объявление, сосед Ивана Петр записывал, какие доски оказывались заклеены этим объявлением. А именно, выяснилось что в i-й день очередное объявление было наклеено таким образом, что занимало доски с l_i-й по r_i-ю включительно. При этом рекламный агент вполне мог заклеить новым объявлением полностью или частично свое же собственное объявление.

Для составления жалобы в администрацию деревни Ивану необходимо удостовериться, что рекламой заклеен весь забор. Помогите ему выяснить, так ли это.

Формат входного файла

В первой строке входного файла даны два натуральных числа m и n — число досок в заборе и число дней, в течение которых вел свои наблюдения Петр ($1 \le m \le 10\,000$, $1 \le n \le 1000$). Далее, в n строках заданы целые числа l_i, r_i ($1 \le l_i \le r_i \le m$), i-я пара чисел описывает отрезок забора, который заклеивались объявлением в i-й день.

Формат выходного файла

Выведите «`YES`», если весь забор был заклеен объявлениями, и «`NO`» в противном случае.

Примеры

checkpaint.in	checkpaint.out
3 3 1 1 2 3 3 3	YES

Задача C. Укрепление мостов

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

Байтландия готовится к военным учениям. Это очень важное мероприятие, даже министр обороны Байтландии контролирует организацию учений на месте. Министр обороны обеспокоен тем, как же пройдут учения танков.

Байтландия состоит из островов, некоторые из которых соединены мостами. Каждый мост соединяет два различных острова, любые два острова соединены напрямую не более чем одним мостом. Байтландцы очень экономный народ, поэтому на каждый остров ведут не более двух мостов.

План мероприятия еще не готов, но известно, что план учений танков будет таким: танки должны будут проехать с одного острова на другой, пользуясь некоторыми мостами, причем не важно, какими именно мостами будут пользоваться танки. В Байтландии много мостов, которые были построены много лет назад и для танков совершенно не предназначены. Поэтому министр обороны решил укрепить некоторые мосты. А конкретно, он хочет укрепить несколько мостов так, чтобы вне зависимости от плана учений, выполнялось условие: если была возможность переехать с острова u на остров v, то после укрепления некоторых мостов можно переехать с острова u на остров v по **укрепленным** мостам. При этом укрепление моста — дорогая операция, поэтому министр хочет укрепить минимальное число мостов.

Министр обороны Байтландии хочет знать, сколько существует различных способов укрепления минимального числа мостов. Два способа считаются различными, если существует мост, который укреплен в одном из способов и не укреплен в другом. Помогите министру обороны найти ответ на волнующий его долгое время вопрос. Поскольку ответ может быть довольно большим, выведите его по модулю $10^9 + 7$.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($1 \le n \le 10^5$, $0 \le m \le 10^5$) — количество островов и количество мостов в Байтландии соответственно.

В следующих m строках заданы мосты, по одному в строке. Каждый мост задан двумя целыми числами: v_i и u_i ($1 \le v_i, u_i \le n$, $v_i \ne u_i$) — номера островов, которые соединяет мост с номером i.

Гарантируется, что каждый мост задан во входном файле не более одного раза.

Гарантируется, что из каждого острова выходят не более чем два моста.

Формат выходного файла

В выходной файл выведите единственное число: остаток от деления количества способов укрепления мостов на число $10^9 + 7$.

Примеры

fortification.in	fortification.out
5 4 1 2 2 3 1 3 4 5	3
2 1 1 2	1

Пояснение

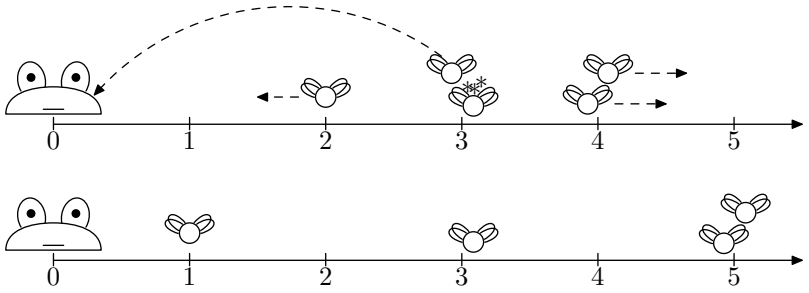
В первом примере существует три способа укрепления мостов: укрепить мосты с номерами $\{1, 2, 4\}$, либо $\{1, 3, 4\}$, либо $\{2, 3, 4\}$.

Задача D. Лягушонок Билли

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

Лягушонок Билли сидел на камне и любовался на закат, когда понял, что проголодался. Он огляделся и с удивлением обнаружил, что в ручье около него копошатся мошки. Ручей представляет собой прямую, на которой расположен и камень, на котором сидит Билли. Лягушонок был очень голоден, и потому захотел съесть всех мошек. У Билли очень длинный язык, поэтому он может, не спрыгивая с камня, съесть любую мошку (но только одну за раз).

Однако высовывать язык на большие расстояния не так-то просто, лягушонок на каждый сантиметр высунутого языка тратит одну единицу энергии. Каждый раз, когда Билли съедает мошку из какой-то точки происходит следующее: все мошки, сидящие слева от съеденной мошки, и все мошки, сидящие справа от нее в ужасе отпрыгивают от места событий на один сантиметр вдоль ручья. Мошки, которые сидят в той же точке, что и съеденная, настолько шокированы этим событием, что не двигаются.



Если мошка в какой-то момент времени прыгает на камень, где сидит Билли, то Билли тут же съедает ее не тратя энергии. При этом другие мошки не перемещаются.

Лягушонок Билли хочет понять — какое минимальное количество единиц энергии ему потребуется для того, чтобы съесть всех мошек. Помогите ему это выяснить.

Формат входного файла

В первой строке входного файла задано одно натуральное числа n ($1 \le n \le 100\,000$) — количество мошек. Во второй строке входного файла задано n натуральных чисел — расстояния каждой из мошек до камня. Известно, что все мошки находятся на одной прямой по одну сторону от камня. Расстояния даны в порядке неубывания. Расстояния не превышают 10^9.

Формат выходного файла

Выведите одно число — минимальное количество единиц энергии, которое потребуется Билли, чтобы съесть всех мошек.

Примеры

frog.in	frog.out
4 2 2 4 4	8

Пояснение

Пояснение к примеру. Сначала Билли съест одну мошку, сидящую в точке 4. Другая мошка, сидящая в этой точке не сдвинется, обе мошки из точки 2 сдвинутся в точку 1. После того, как он съест вторую мошку в точке 4, обе мошки из точки 1 отпрыгнут в точку 0, где и будут сразу съедены.

Задача E. Половина

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

У доброжелательного Даниила есть несколько яблок. В силу своей природной доброжелательности, каждый раз, когда он встречает какого-либо своего друга, он смотрит на яблоки, которые у него есть и отдает другу половину.

Но Даниил не одинаково любит всех своих друзей, поэтому некоторым из них он отдает половину яблока, а некоторым — половину имеющихся у него яблок. При этом с глазомером у Даниила не так хорошо, как со щедростью, и делить яблоки более, чем на две части, у него не получается. Поэтому, если он встречает друга, а у него нецелое число яблок, то он вынужден отдать половину яблока.

Утром у Даниила было n яблок, а за день Даниил встретил k друзей. Выясните, сколько яблок у него могло остаться вечером.

Формат входного файла

Входной файл содержит два целых числа: n — количество яблок у Даниила и k — количество встреченных им за день друзей ($1 \le n \le 1000$, $1 \le k \le 1000$).

Формат выходного файла

Первая строка выходного файла должна содержать число m — количество вариантов ответа на вопрос, сколько яблок может быть у Даниила вечером. Следующая строка должна содержать m вещественных чисел, отсортированных по возрастанию — варианты ответов.

Примеры

half.in	half.out
6 1	2 3.0 5.5

Задача F. Инициализация массива

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

Во многих языках программирования есть функции, которые отвечают за заполнение всего массива или некоторой его части определенным значением. В языке Pascal это функция `fillchar()`, в Java — `Arrays.fill()`, в C++ — `memset()`. В новом языке программирования J# появилась функция `mark()`, которая умеет работать только с массивами логического типа.

Функция `mark`, вызванная от двух параметров a и b, присваивает всем элементам массива с индексами от a до b включительно значение `true`, Так, если взять массив длины 4, элементы которого нумеруются с единицы и все значения в котором изначально равны `false`, и выполнить с ним операции `mark(1, 3)` и `mark(2, 4)`, то весь массив окажется заполнен значениями `true`.

Одним из первых заданий для тех, кто начинает изучать J#, является написание программы, содержащей ровно M операций `mark`, и полностью заполняющей значениями `true` массив длины N, изначально заполненный значениями `false`.

Вы быстро справились с этим заданием, и теперь задумались: сколькими различными способами это можно сделать? Различными считаются такие способы, в которых i-я операция `mark` в двух программах запущена с разными параметрами хотя бы для одного i от 1 до M. Это число может быть большим, поэтому требуется посчитать его по модулю $10^9 + 7$.

Формат входного файла

В первой строке входного файла даны два натуральных числа N и M — длина массива и количество операций `mark`, которые должны быть в программе. ($1 \le N, M \le 70$).

Формат выходного файла

В единственной строке выходного файла выведите остаток от деления числа способов заполнить массив из N элементов значениями `true` с помощью M вызовов операции `mark` на число $10^9 + 7$.

Примеры

init.in	init.out
2 2	7

Пояснение

Искомые варианты:

- `mark(1, 1); mark(1, 2)`
- `mark(1, 1); mark(2, 2)`
- `mark(1, 2); mark(1, 1)`
- `mark(1, 2); mark(1, 2)`
- `mark(1, 2); mark(2, 2)`
- `mark(2, 2); mark(1, 1)`
- `mark(2, 2); mark(1, 2)`

Задача G. Обработка строки

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

В новых процессорах, проектируемых корпорацией MBI, для работы со строками выделен отдельный сопроцессор. Этому сопроцессору можно передать строку, состоящую из строчных символов латинского алфавита, и программу, описывающую обработку строки. После этого процессор задумается и, через долю секунды, вернет результат обработки строки.

Работа еще не завершена, поэтому пока что он умеет обрабатывать строки только одним способом. Строка, которую ему передают на ввод, должна состоять из 2^k символов, а программа обработки этой строки — из $2^k - 1$ чисел, каждое из которых является нулем или единицей.

Если строка, поданная на вход сопроцессору, состоит из одного символа, то результат обработки равен этой же строке. Если строка длиннее, то выполняются следующие действия.

Сначала он разобьет строку на две части равной длины, а программу — на три части. При этом длины первых двух частей программы будут равны между собой, а третьей частью будет последнее число программы. После этого первая половина строки превратится в результат ее обработки первой частью программы, а вторая — в результат ее обработки второй частью программы. После этого, если оставшееся число равно нулю, вторая половина строки будет дописана справа к первой, если же оно равно единице — первая половина будет дописана справа ко второй.

Так, например, если на вход сопроцессору подана строка «`abcd`» и программа $(0, 1, 1)$, то результатом обработки будет строка «`dcab`».

Вам поступило предложение принять участие в его тестировании. Отказаться от такого лестного предложения вы не смогли, поэтому теперь вам необходимо научиться составлять программу для сопроцессора, с помощью которой он преобразует строку S в строку T, или же определять, что такой программы не существует.

Формат входного файла

Первая строка входного файла содержит одно целое число n — количество случаев, которые вам необходимо разобрать. В следующих строках описаны сами n случаев.

Первая строка описания очередного случая содержит строку S, состоящую только из строчных букв латинского алфавита. Длина строки S равна 2^k, где $1 \le k \le 16$. Вторая строка описания очередного случая содержит строку T, также состоящую только из строчных букв латинского алфавита. Длина строки T равна длине строки S.

Суммарная длина всех строк S в одном входном файле не превышает 100 000.

Формат выходного файла

Для каждого случая в очередной строке выходного файла слово «`Yes`», если программа для сопроцессора, преобразовывающая строку S в строку T существует, и «`No`» в противном случае.

Если ответ «`Yes`», в следующей строке выведите $2^k - 1$ чисел, разделенных пробелами — программу, которую для этого преобразования необходимо передать сопроцессору. Если таких программ несколько — выведите любую.

Примеры

processing.in	processing.out
2 abacabab baababca abacabab bbaaabca	Yes 0 1 0 1 0 0 1 No

Задача H. Ребус

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

На прошлом уроке английского Пете задали домашнее задание. Оно заключалось в разгадывании ребусов. Каждый ребус — это последовательность картинок. С двух сторон от каждой картинки могут располагаться апострофы. Каждая картинка обозначает некоторое слово. Предположим, что перед некоторой картинкой нарисовано i апострофов, а после нее j апострофов. Это значит, что у слова, которое сопоставляется картинке, необходимо убрать i букв с начала и j с конца, а оставшуюся его часть записать вместо картинки и апострофов. Так необходимо сделать с каждой картинкой и окружающими ее апострофами. После этого нужно «склеить» получившиеся кусочки в одно слово. Оно и будет разгадкой ребуса.

У Пети нет проблем с тем, чтобы сопоставить каждой картинке слово. Но ему очень лень заниматься отбрасыванием лишних букв и склеиванием слов. Поэтому он попросил вас помочь ему. Дана строка, которая состоит из маленьких латинских букв, апострофов (символов с кодом 39), а также пробелов, которые разделяют слова. Апостроф относиться к некоторому слову, если между ними нет пробела. Если апостроф стоит слева от слова, то у него необходимо убрать одну букву с начала, если справа — с конца. Потом необходимо склеить все слова в одно.

Например, пусть дана строчка «team ’’’’school ’’olympiad’’’». В первом слове ничего изменять не надо, так как к нему не относится ни один апостроф. Во втором необходимо убрать первые четыре буквы и получить «ol», из третьего слова получится «ymp». После скеливания трех кусочков получим строку «teamolymp».

Формат входного файла

В первой строке входного файла дан ребус, длиной не более 100 символов, который необходимо решить. Гарантируется, что в строчке присутствуют только апострофы (код символа — 39), пробелы и маленькие латинские буквы, а также, что ребус корректен — нет слова, у которого нобходимо удалить букв больше, чем его длина.

Формат выходного файла

Выведите одно слово — ответ на ребус.

Примеры

rebus.in	rebus.out
team ’’’’school ’’olympiad’’’	teamolymp

Задача I. Племя тив

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

Каждый год профессор Иванов ездит в Африку с целью изучить племена, которые там проживают. В этом году он ездил в гости к племени тив. Профессор довольно быстро научился понимать их язык, выучил многие их обряды, однако, он никак не мог понять записанные цифрами тив числа. Как и мы, члены племени используют позиционную систему счисления с основанием 10. Но цифры в племени тив обозначают символами, не похожими на обычные цифры от 0 до 9.

Профессор обозначил эти символы буквами от ‘a’ до ‘j’, но не может понять, какой цифре соответствует какой символ. Тогда вождь племени дал ему список из n неотрицательных чисел, записанных без ведущих нулей, и сказал, что числа в нем отсортированы строго по возрастанию.

Помогите профессору восстановить по этому списку какое-нибудь соответствие символов цифрам.

Формат входного файла

В первой строке входного файла дано одно натуральное числа n ($2 \le n \le 10$) — количество слов в списке. Следующие n строк содержат выданные вождем числа племени тив, по одному числу в строке. Длина каждого числа не превышает 9.

Формат выходного файла

В первой строке файла выведите «`Yes`», если ответ существует, в этом случае в следующей строке выведите цифры, которые соответствуют символам, обозначенным ‘a’..‘j’, в этом порядке. Если существует несколько ответов, то выведете любой из них.

Если профессор понял что-то неправильно, и ответа не существует, выведете «`No`».

Примеры

tiv.in	tiv.out
4 a da dd cc	Yes 1 0 3 2 4 5 6 7 8 9
4 a j jb ac	No

Задача J. День рождения викинга

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

Немногие знают, но помимо боевого и трудового топора викинги используют кухонный топор. Сегодня у Хильдсвегсамара Харфагра по прозвищу Кровавый Топор день рождения. Хильдсвегсамар, как всегда, зашел к своей бабушке Одаудлегье подкрепиться и обнаружил, что бабушка испекла огромный круглый торт и положила его на стол. Теперь его необходимо разделить на две части для Хильдсвегсамара и его бабушки.

В семье Харфагров принято делить торты одним ударом топора. После удара топора образуется разрез в том месте, где ударил викинг. Разрез представляет собой отрезок, длина которого не больше длины лезвия топора. Торт оказывается поделен на две части, если разрез соединяет две точки на границе торта. Так как длина топора фиксирована, возможно, торт не удастся разделить на две равные части за один разрез. Поэтому викинги хотят разделить одним разрезом торт так, чтобы им достались по возможности наиболее близкие по площади части.

Хильдсвегсамар быстро догадался, что торт нужно разделить по хорде максимальной длины, не превосходящей длины лезвия, но не может найти где именно нужно резать. Введем на столе прямоугольную декартову систему координат с центром, совпадающим с центром торта. Помогите Хильдсвегсамару найти две точки на границе торта, через которые должен проходить разрез, чтобы разделить торт наиболее честно.

Формат входного файла

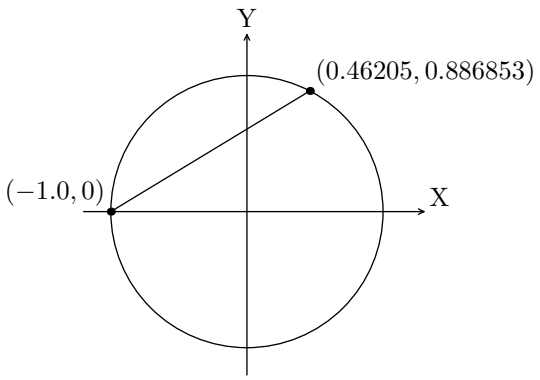
В первой строке входного файла даны два вещественных числа R и L — радиус торта и длина лезвия соответственно с не более чем тремя знаками после десятичной точки ($1 \le R, L \le 1000$).

Формат выходного файла

Первая строка выходного файла должна содержать координаты первой точки на границе, вторая строка координаты второй точки. Расстояние от точек до центра должно отличаться от R не более чем на 10^{-6}. Расстояние между точками должно отличаться от максимально возможной хорды не превосходящей L не более чем на 10^{-6}.

Примеры

vikings.in	vikings.out
1.0 2.0	0 -1.0 0 1.0
1.0 1.71	-1.0 0 0.46205 0.886853



Одно из возможных расположений разреза во втором примере.