

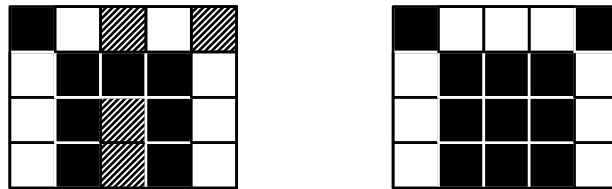
Задача А. Шахматный детектив

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

Известный частный сыщик поставил чашку с чаем на специальную подогревающую подставку, питающуюся от USB-порта его компьютера, и приступил к обдумыванию очередного запутанного преступления. Через пару часов раздумий он понял, что для разгадки этого дела достаточно определить, была ли на месте преступления шахматная доска.

Недавно он получил по электронной почте фотографию места преступления. Подозрительный фрагмент (тот, на котором изображен предмет, похожий на шахматную доску) уже был скопирован в отдельный файл, но вдруг выяснилось, что, поскольку фотография была сжата с потерей качества, некоторые пиксели на ней из белых или черных стали серыми. Таким образом, определение того, является ли сфотографированный предмет шахматной доской, стало намного более сложным.

Все усложняется тем, что на фотографию могла попасть не вся шахматная доска, а лишь ее часть. Например, на приведенном рисунке на фотографию попала часть доски, у которой каждое поле имеет длину стороны, равную трем пикселям.



Помогите частному сыщику в расследовании преступления. Напишите программу, которая определит, может ли заданный фрагмент фотографии быть изображением части шахматной доски, и, если может, восстановит изображение шахматной доски до сжатия.

Шахматная доска — это квадрат, разбитый на x^2 (для некоторого x) равных квадратов — полей. Стороны полей параллельны сторонам изображения. Длина стороны каждого поля шахматной доски выражается целым числом пикселей. Все пиксели, принадлежащие одному полю, покрашены в один и тот же цвет — черный или белый. При этом соседние поля (поля, имеющие общую сторону) покрашены в различные цвета.

Формат входного файла

Первая строка входного файла содержит два целых числа: m и n — размеры фрагмента фотографии в пикселях ($1 \leq m, n \leq 250$).

Следующие m строк содержат по n символов каждая, j -й символ i -й строки соответствует пикселю с координатами (i, j) . Символ «.» (точка) означает белый пиксель, символ «*» — черный, символ «?» — серый.

Формат выходного файла

Если заданный фрагмент фотографии может быть изображением части шахматной доски, выведите в выходной файл слово «YES». После этого выведите m строк по n символов в каждой. Они должны содержать изображение соответствующей части шахматной доски в том же формате, что и во входном файле, только серые пиксели должны быть заменены на белые или черные. Если решений несколько, выведите любое.

В противном случае выведите в выходной файл слово «NO».

Пример

<code>chess.in</code>	<code>chess.out</code>
<code>4 5 * . ? . ? . * * * . . * ? * . . * ? * .</code>	<code>YES * . . . * . * * * . . * * * . . * * * .</code>
<code>4 5 . . ? . . . * * * . . * ? * . . * ? * .</code>	<code>NO</code>

Задача В. Кинотеатр

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

Марья Ивановна с Марьей Михайловной привели школьников в кинотеатр. Чтобы не было никаких обид, Марья Ивановна построила всех школьников по алфавиту и рассадила их: сначала в первый ряд слева направо, затем во второй слева направо и т.д., заполнив весь зал из n рядов по m кресел. Тут пришла Марья Михайловна и сказала, что ребята сели неправильно — надо пересесть. Она предложила сначала заполнить все первые места от первого ряда к последнему, затем все вторые места и т. д.

Определите, сколько школьников после такой пересадки останется на своем месте.

Например, если $n = 3$ и $m = 3$, то в первом случае дети сядут так:

```
1 2 3
4 5 6
7 8 9
```

а во втором — так:

```
1 4 7
2 5 8
3 6 9
```

Таким образом, три школьника: 1, 5 и 9 останутся на своих местах.

Формат входного файла

Входной файл содержит два целых числа n и m ($1 \leq n, m \leq 10^9$).

Формат выходного файла

Выведите количество школьников, которые останутся на своих местах.

Примеры

<code>cinema.in</code>	<code>cinema.out</code>
3 3	3
2 4	2

Задача С. Цифровое табло

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

В комнате у Аркадия Семеновича Тапкина стоят электронные часы. Цифры на этих часах показываются в специальной псевдографике. А именно каждое поле, на котором изображается цифра, состоит из w ячеек в ширину и h ячеек в высоту (при этом ячейки на поле имеют форму квадратов).

Но недавно у Аркадия Семеновича появилась проблема. Последнее время он стал плохо видеть. В связи с этим он хочет увеличить изображение этих цифр. Он уже приладил старый 19" монитор к часам, и теперь дело осталось за малым. Осталось написать программу, которая будет рисовать цифры на дисплее. Аркадий Семенович хочет увеличить изображение в k раз и сделать толщину линий равной d . Помогите ему в этом.

Опишем более формально понятие «увеличить в k раз». Занумеруем ячейки поля $w \times h$ сверху вниз и слева направо. Таким образом, верхняя левая ячейка имеет координаты $(0, 0)$, правая нижняя — $(w-1, h-1)$, правая верхняя — $(w-1, 0)$, левая нижняя — $(0, h-1)$. Кроме этого, введем декартову прямоугольную систему координат так, что начало координат находится в центре верхней левой ячейки, ось Ox направлена вправо, ось Oy — вниз, длину единичного отрезка примем равной длине стороны ячейки. Таким образом, координаты центра ячейки совпадают с ее координатами во введенной нумерации.

Каждая десятичная цифра задается набором составляющих ее изображение отрезков. Для простоты каждый из отрезков либо параллелен одной из координатных осей, либо идет под углом в 45 градусов к ней.

Увеличенная в k раз цифра рисуется на поле размером $(w-1) \cdot (k-1) + w$ ячеек по горизонтали на $(h-1) \cdot (k-1) + h$ ячеек по вертикали.

При увеличении некоторой цифры в k раз производятся следующие операции. Координаты точек, являющихся концами отрезков, составляющих цифру, умножаются на k . После этого закрашиваются те ячейки, через центры которых проходят эти отрезки. Эти ячейки будем называть *основными*.

После этого, для того, чтобы получить толщину линий равную d , дополнительно закрашиваются те ячейки, центры которых располагаются на расстоянии, не превышающем $(d-1)$ от центров основных ячеек. *Расстоянием* между точками $A(x_A, y_A)$ и $B(x_B, y_B)$ будем называть число $\rho(A, B) = |x_A - x_B| + |y_A - y_B|$.

По описанию цифры и параметрам k и d выведите изображение цифры, увеличенное в k раз, с толщиной линий d .

Формат входного файла

Первая строка входного файла содержит целые числа k и d ($1 \leq k \leq 100$, $1 \leq d \leq 500$). Вторая строка входного файла содержит целые числа w и h ($1 \leq w, h \leq 10$).

Третья строка входного файла содержит целое число n ($1 \leq n \leq 100$) — количество отрезков в описании цифры. Далее следуют n строк, каждая из которых описывает один отрезок. Описание отрезка состоит из четырех целых чисел: x_1, y_1, x_2, y_2 ($0 \leq x_1, x_2 < w$, $0 \leq y_1, y_2 < h$) — координат концов отрезка.

Каждый из отрезков либо параллелен одной из координатных осей, либо идет под углом в 45 градусов к ней. Все отрезки имеют ненулевую длину.

Формат выходного файла

Выходной файл должен содержать ровно $(h-1) \cdot (k-1) + h$ строк по $(w-1) \cdot (k-1) + w$ символов в каждой, j -ый символ i -ой строки должен быть равен символу «*» (звездочка), если ячейка с центром в точке (j, i) закрашена, и символу «.» (точка) — иначе.

Задача D. Выборы

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

В одной демократической стране приближаются парламентские выборы. Выборы проходят по следующей схеме: каждый житель страны, достигший восемнадцатилетнего возраста, отдает свой голос за одну из политических партий. После этого партия, которая набрала максимальное количество голосов, считается победившей на выборах и формирует правительство. Если несколько партий набрали одинаковое максимальное количество голосов, то они должны сформировать коалиционное правительство, что обычно приводит к длительным переговорам.

Один бизнесмен решил выгодно вложить свои средства, и собрался поддержать на выборах некоторые партии. В результате поддержки он планирует добиться победы одной из этих партий, которая затем сформирует правительство, которое будет действовать в его интересах. При этом возможность формирования коалиционного правительства его не устраивает, поэтому он планирует добиться строгой победы одной из партий.

Чтобы повлиять на исход выборов, бизнесмен собирается выделить деньги на агитационную работу среди жителей страны. Исследование рынка показало, что для того чтобы один житель сменил свои политические воззрения, требуется потратить одну условную единицу. Кроме того, чтобы i -я партия в случае победы сформировала правительство, которое будет действовать в интересах бизнесмена, необходимо дать лидеру этой партии взятку в размере p_i условных единиц. При этом некоторые партии оказались идеологически устойчивыми и не согласны на сотрудничество с бизнесменом ни за какие деньги.

По результатам последних опросов известно, сколько граждан планируют проголосовать за каждую партию перед началом агитационной компании. Помогите бизнесмену выбрать, какую партию следует подкупить, и какое количество граждан придется убедить сменить свои политические воззрения, чтобы выбранная партия победила, учитывая, что бизнесмен хочет потратить на всю операцию минимальное количество денег.

Формат входного файла

Первая строка входного файла содержит целое число n — количество партий ($1 \leq n \leq 10^5$). Следующие n строк описывают партии. Каждая из этих строк содержит по два целых числа: v_i — количество жителей, которые собираются проголосовать за эту партию перед началом агитационной компании, и p_i — взятка, которую необходимо дать лидеру партии для того, чтобы сформированное ей в случае победы правительство действовало в интересах бизнесмена ($1 \leq v_i \leq 10^6$, $1 \leq p_i \leq 10^6$ или $p_i = -1$). Если партия является идеологически устойчивой, то p_i равно -1 . Гарантируется, что хотя бы одно p_i не равно -1 .

Формат выходного файла

На первой строке выходного файла выведите минимальную сумму, которую придется потратить бизнесмену. На второй строке выведите номер партии, лидеру которой следует дать взятку. На третьей строке выведите n целых чисел — количество голосов, которые будут отданы за каждую из партий после осуществления операции. Если оптимальных решений несколько, выведите любое.

Примеры

<code>elect.in</code>	<code>elect.out</code>
3	6
7 -1	3
2 8	3 3 4
1 2	

Задача Е. Треугольная рамка

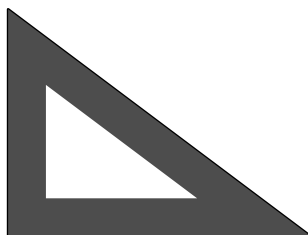
Имя входного файла: `frame.in`
Имя выходного файла: `frame.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Картины художников-абстракционистов весьма необычны. На них часто изображены абсолютно непонятные предметы. Один известный художник-абстракционист решил добавить своим картинам оригинальности следующим способом. Вместо прямоугольных рамок, в которые обычно вставляются картины, он решил на ближайшей выставке использовать треугольные.

Однако все оказалось совсем не так просто. Художник изготовил рамку, поместил в нее картину и понял, что что-то не так. Рамка получилась слишком широкой, и картина выглядела совсем не так ярко, как он ожидал.

Немного поразмыслив, художник понял, что то, насколько рамка «подходит» для картины, определяется площадью рамки. Кроме этого он понял, что рамки надо не изготавливать самостоятельно, а покупать в специальном магазине. Заглянув в прайс-лист магазина, он увидел, что для каждой рамки в нем указаны длины внешних сторон и ширина.

Поясним подробнее то, как выглядит треугольная рамка. Ее изготовление происходит следующим образом: берется доска из красного дерева, имеющая форму треугольника со сторонами a , b и c . После этого стороны этого треугольника мысленно сдвигаются внутрь него на расстояние d (измеряемое по перпендикуляру к соответствующей стороне). На точках пересечения «сдвинутых» сторон строится маленький треугольник, который затем вырезается из исходного. Пример рамки со сторонами $a = 6$, $b = 8$, $c = 10$ и шириной $d = 1$ показан на рисунке.



Помогите художнику по имеющимся в прайс-листе данным вычислить площадь рамки.

Формат входного файла

Входной файл содержит четыре целых числа a , b , c , d ($1 \leq a, b, c, d \leq 1000$) — длины внешних сторон рамки и ее ширину, соответственно. Гарантируется, что треугольник со сторонами a , b и c существует, и что в треугольнике есть точка, расстояние от которой до ближайшей стороны строго больше d .

Формат выходного файла

В выходной файл выведите площадь рамки с точностью не меньше 10^{-5} .

Примеры

<code>frame.in</code>	<code>frame.out</code>
6 8 10 1	18.00000

Задача F. Форматирование документа

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

Вася пишет новую версию своего офисного пакета «Closed Office». Недавно он начал работу над редактором «Dword», входящим в состав пакета.

Последняя проблема, с которой столкнулся Вася – размещение рисунков в документе. Он никак не может добиться стабильного отображения рисунков в тех местах, в которые он их помещает. Окончательно отчаявшись написать соответствующий модуль самостоятельно, Вася решил обратиться за помощью к вам. Напишите программу, которая будет осуществлять размещение документа на странице.

Документ в формате редактора «Dword» представляет собой последовательность абзацев. Каждый абзац представляет собой последовательность элементов – слов и рисунков. Элементы одного абзаца разделены пробелами и/или переводом строки. Абзацы разделены пустой строкой. Строка, состоящая только из пробелов, считается пустой.

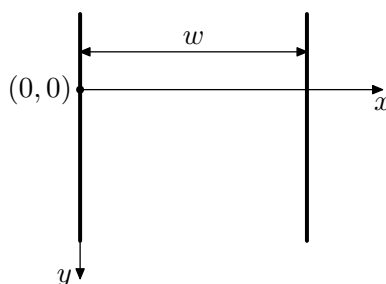
Слово – это последовательность символов, состоящая из букв латинского алфавита, цифр, и знаков препинания: «.», «,», «:», «;», «!», «?», «-», «'».

Рисунок описывается следующим образом: «(image *параметры рисунка*)». Каждый параметр рисунка имеет вид «*имя=значение*». Параметры рисунка разделены пробелами и/или переводом строки. У каждого рисунка обязательно есть следующие параметры:

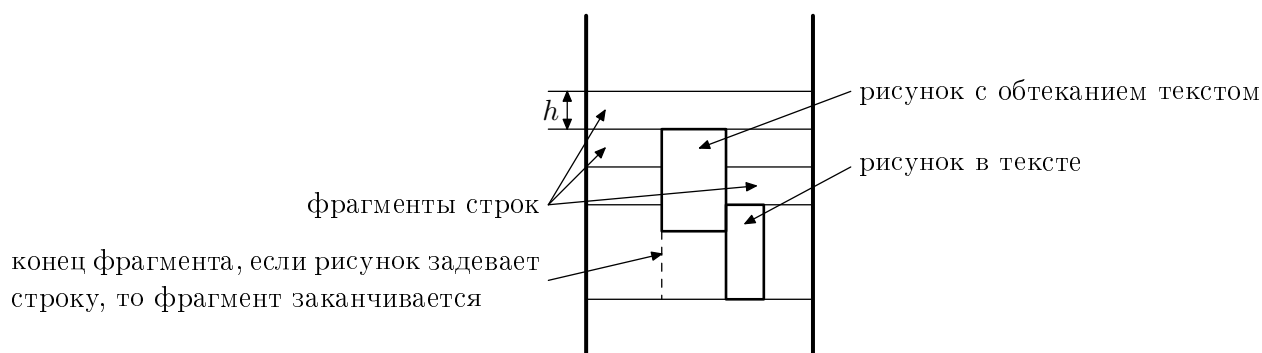
Параметр	Описание
width	Целое положительное число – ширина рисунка в пикселях
height	Целое положительное число – высота рисунка в пикселях
layout	Одно из следующих значений: embedded (в тексте), surrounded (обтекание текстом), floating (свободное) – описывает расположение рисунка относительно текста

Документ размещается на бесконечной вверх и вниз странице шириной w пикселей (разбиение на конечные по высоте страницы планируется в следующей версии редактора). Одна из точек на левой границе страницы условно считается точкой с ординатой равной нулю. Ордината увеличивается вниз.

Размещение документа происходит следующим образом. Абзацы размещаются по очереди. Первый абзац размещается так, что его верхняя граница имеет ординату 0.



Абзац размещается следующим образом. Элементы располагаются по строкам. Каждая строка исходно имеет высоту h пикселей. В процессе размещения рисунков высота строк может увеличиваться, и строки могут разбиваться рисунками на фрагменты.



Слова размещаются следующим образом. Считается, что каждый символ имеет ширину c пикселей. Перед каждым словом, кроме первого во фрагменте, ставится пробел шириной также в c пикселей. Если слово помещается в текущем фрагменте, то оно размещается на нем. Если слово не помещается в текущем фрагменте, то оно размещается в первом фрагменте текущей строки, расположенном правее текущего, в котором оно помещается. Если такого фрагмента нет, то начинается новая строка, и поиск подходящего фрагмента продолжается в ней. Слово всегда «прижимается» к верхней границе строки.

Размещение рисунка зависит от его расположения относительно текста.

Если расположение рисунка относительно текста установлено в «*embedded*», то он располагается так же как слово, за тем исключением, что его ширина равна ширине, указанной в параметрах рисунка. Кроме того, если высота рисунка больше текущей высоты строки, то она увеличивается до высоты рисунка (при этом верхняя граница строки не перемещается, а смещается вниз нижняя граница). Если рисунок типа «*embedded*» не первый элемент во фрагменте, то перед ним ставится пробел шириной c пикселей. Рисунки типа «*embedded*» также прижимаются к верхней границе строки.

Если расположение рисунка относительно текста установлено в «*surrounded*», то рисунок размещается следующим образом. Сначала аналогично находится первый фрагмент, в котором рисунок помещается по ширине. При этом перед рисунком этого типа не ставится пробел, даже если это не первый элемент во фрагменте.

После этого рисунок размещается следующим образом: верхний край рисунка совпадает с верхней границей строки, в которой находится найденный фрагмент, а сам рисунок продолжается вниз. При этом строки, через которые он проходит, разбиваются им на фрагменты.

Если расположение рисунка относительно текста установлено в «*floating*», то рисунок размещается поверх текста и других рисунков и никак с ними не взаимодействует. В этом случае у рисунка есть два дополнительных параметра: «*dx*» и «*dy*» — целые числа, задающие смещение в пикселях верхнего левого угла рисунка вправо и вниз, соответственно, относительно позиции, где находится верхний правый угол предыдущего слова или рисунка (или самой левой верхней точки первой строки абзаца, если рисунок — первый элемент абзаца).

Если при размещении рисунка таким образом он выходит за левую границу страницы, то он смещается вправо, так, чтобы его левый край совпадал с левой границей страницы. Аналогично, если рисунок выходит за правую границу страницы, то он смещается влево, чтобы его правый край совпадал с правой границей страницы.

Верхняя граница следующего абзаца совпадает с более низкой точкой из нижней границы последней строки и самой нижней границы рисунков типа «*surrounded*» предыдущего абзаца.

По заданным w , h , c и документу найдите координаты верхних левых углов всех рисунков в документе.

Формат входного файла

Первая строка входного файла содержит три целых числа: w , h и c ($1 \leq w \leq 1000$, $1 \leq h \leq 50$, $1 \leq c \leq w$).

Далее следует документ. Размер входного файла не превышает 1000 байт. Гарантируется, что ширина любого слова и любого рисунка не превышает w . Высота всех рисунков не пре-

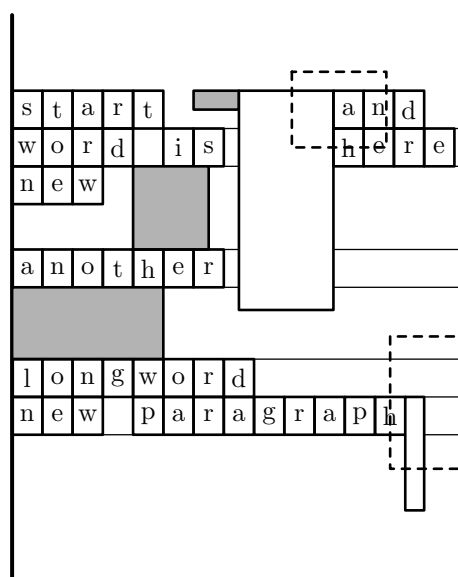
вышает 1000. Относительное смещение всех рисунков типа «floating» не превышает 1000 по абсолютной величине.

Формат выходного файла

Выведите в выходной файл по два числа для каждого рисунка – координаты его верхнего левого угла. Выводите координаты рисунков в том порядке, в котором они встречаются во входном файле.

Примеры

layout.in
120 10 8 start (image layout=embedded width=12 height=5) (image layout=surrounded width=25 height=58) and word is (image layout=floating dx=18 dy=-15 width=25 height=20) here new (image layout=embedded width=20 height=22) another (image layout=embedded width=40 height=19) longword new paragraph (image layout=surrounded width=5 height=30) (image layout=floating width=20 height=35 dx=50 dy=-16)
layout.out
48 0 60 0 74 -5 32 20 0 52 104 81 100 65



Задача G. Максимальная сумма

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

Сегодня на страницах газеты «Математический досуг» была опубликована необычная математическая головоломка. Одна из страниц газеты полностью занята прямоугольной таблицей, состоящей из m строк и n столбцов. В каждой ячейке таблицы записано некоторое целое число.

Для решения головоломки требуется найти такой невырожденный прямоугольник с вершинами в центрах ячеек таблицы, и сторонами, параллельными сторонам таблицы, чтобы сумма чисел, записанных в ячейках на границе получившегося прямоугольника, была максимальной.

1	-2	-1	3
-10	-5	1	-4
1	-1	2	-2
3	0	0	-1
2	2	-1	2

Безуспешно потратив несколько часов на решение головоломки, Саша решил написать программу, которая сделала бы это за него. Но и тут его постигла неудача. Теперь ему ничего не остается, как обратиться за помощью к вам.

Напишите программу, которая по заданной таблице найдет искомый прямоугольник.

Формат входного файла

На первой строке входного файла записаны два целых числа m и n ($2 \leq m, n \leq 300$). Далее следует описание таблицы — m строк, каждая из которых содержит по n целых чисел $a_{i,j}$ ($-10^4 \leq a_{i,j} \leq 10^4$).

Формат выходного файла

На первой строке выходного файла выведите целое число s — максимальную сумму чисел на границе искомого прямоугольника. На второй строке выведите четыре натуральных числа: x_1, y_1, x_2, y_2 — координаты левой верхней и правой нижней ячейки выбранного прямоугольника, соответственно (здесь x — номер строки, а y — номер столбца, строки нумеруются сверху вниз, начиная с единицы, столбцы нумеруются слева направо, начиная с единицы). Если оптимальных решений несколько, выведите любое.

Примеры

<code>max.in</code>	<code>max.out</code>
2 3 1 1 1 1 1 1	6 1 1 2 3
5 4 9 -2 -1 3 -10 -5 1 -4 1 -1 2 -2 3 0 0 -1 2 2 -1 2	8 3 1 5 3

Задача Н. Метро

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

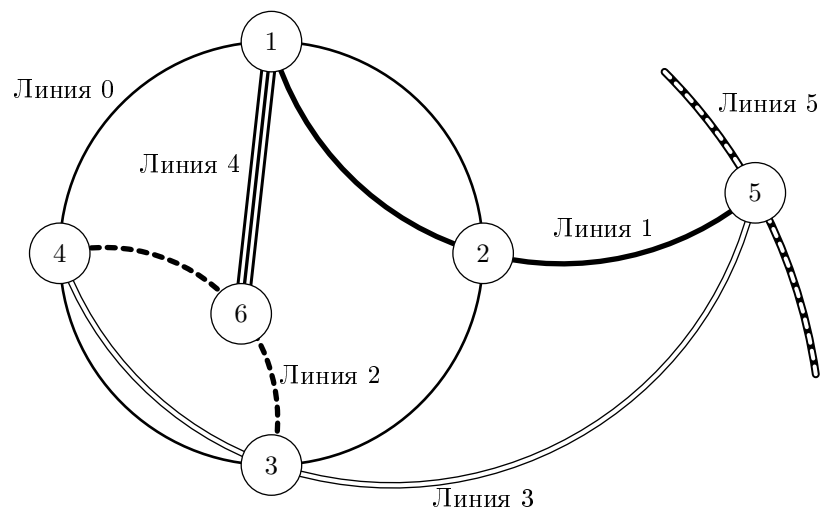
Женя получил письмо от Леша со словесным описанием схемы метро в его городе. Метро содержит одну кольцевую линию. Каждая из остальных линий пересекается с кольцевой не более чем в двух местах, причем в точках пересечения организованы пересадочные станции. В одном месте кольцевую линию могут пересекать сразу несколько линий, имеющих общую пересадочную станцию.

Кроме этих пересадочных станций каждая из линий имеет не более одной пересадочной станции для перехода на другие, отличные от кольцевой, линии. На такой станции также может быть организована пересадка сразу на несколько линий.

Для каждой пересадочной станции Леша описал, какие линии на ней пересекаются, и указал порядок расположения пересадочных станций на кольцевой линии. Он утверждает, что все линии расположены на одной глубине и других пересечений, помимо пересадочных узлов, не имеют. Чтобы проверить это утверждение, Женя стал по словесному описанию рисовать схему метро, но у него не получилось.

Помогите Жене написать программу, которая будет проверять, действительно ли может существовать схема метро, соответствующая полученному описанию.

На рисунке приведена возможная схема метро, соответствующая второму примеру.



Формат входного файла

В первой строке входного файла содержится число k — количество линий метро в городе ($1 \leq k \leq 10$). Все линии пронумерованы от 0 до $k-1$, кольцевая линия имеет номер 0. Во второй строке записано число n — количество пересадочных станций. Каждая из следующих n строк описывает линии, пересекающиеся на соответствующей пересадочной станции, причем сначала следуют описания пересадочных станций, расположенных на кольцевой линии, в порядке их расположения на ней. Описание каждого узла начинается с количества пересекающихся в нем линий, затем следуют номера линий.

Формат выходного файла

Выведите в выходной файл слово YES, если по описанию можно построить схему метро, и NO в противном случае.

Примеры

metro.in	metro.out
4 6 2 0 1 2 0 2 2 0 3 2 0 1 2 0 2 2 0 3	NO
6 6 3 0 1 4 2 0 1 3 0 2 3 3 0 2 3 3 1 3 5 2 2 4	YES

Задача I. Число

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

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрежала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

Формат входного файла

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходного файла

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

<code>number.in</code>	<code>number.out</code>
2 20 004 66	66220004
3	3

Задача J. Переключение между окнами

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

Когда пользователь работает в операционной системе Windows, у него часто запущено несколько приложений. Каждое из приложений работает в отдельном окне. Для переключения между окнами используется комбинация клавиш «Alt+Tab». Эта комбинация делает активным окно, в котором пользователь работал перед тем, как перейти в текущее активное окно.

Чтобы переключиться в другое окно, можно нажать клавишу «Alt» и затем, не отпуская ее, несколько раз нажать клавишу «Tab». Чтобы понять, какое окно станет активным после этого, воспользуемся следующей моделью. Пусть запущено n приложений. Приложения в операционной системе организованы в виде списка и упорядочены по убыванию времени последней активности. То есть приложение, окно которого является активным в настоящий момент — первое в списке, приложение, окно которого было активно перед этим — второе, и т. д.

Если нажать клавишу «Alt» и затем, не отпуская ее, нажать клавишу «Tab» k раз, то активным станет окно приложения, которое находится на $(k \bmod n) + 1$ -м месте в списке. Здесь $a \bmod b$ означает остаток от деления a на b . Иными словами, операционная система рассматривает список как циклический, переходя после последнего элемента списка к первому.

При запуске нового приложения оно добавляется в начало списка.

Задана последовательность действий пользователя, где каждое действие — либо запуск приложения, либо переключение между окнами. Выведите в выходной файл, в каком порядке пользователь работал с приложениями.

Формат входного файла

Первая строка входного файла содержит целое число n — количество действий пользователя ($1 \leq n \leq 1000$). Следующие n строк содержат описание действий пользователя.

Запуск приложения описывается строкой «Run <имя приложения>». Здесь «<имя приложения>» — строка из не более чем 100 латинских букв, цифр и пробелов. Она отделена от слова «Run» ровно одним пробелом. Все имена приложений различны. Большие и маленькие буквы считаются различными.

Переключение между приложениями описывается строкой «Alt+Tab+...+Tab», здесь подстрока «+Tab» повторена в точности столько раз, сколько раз пользователь нажал клавишу «Tab», не отпуская клавишу «Alt». Это количество не превышает 100.

Первая команда во входном файле — всегда команда «Run».

Формат выходного файла

Выведите в выходной файл n строк — последовательность имен приложений, с которыми работал пользователь в порядке, в котором их окна становились активными.

Примеры

windows.in	windows.out
6	Mozilla Firefox
Run Mozilla Firefox	Free Pascal
Run Free Pascal	Mozilla Firefox
Alt+Tab	Miranda IM
Run Miranda IM	Free Pascal
Alt+Tab+Tab	Free Pascal
Alt+Tab+Tab+Tab	