

Задача А Перегоны

Имя входного файла: a.in
 Имя выходного файла: a.out
 Максимальное время работы на одном тесте: 3 секунды
 Максимальный объем используемой памяти: 8 мегабайт

На некоторой железнодорожной ветке расположено N станций, которые последовательно пронумерованы числами от 1 до N . Известны расстояния между некоторыми станциями. Требуется точно вычислить длины всех перегонов между соседними станциями или указать, что это сделать невозможно (то есть приведенная информация является противоречивой или ее недостаточно).

Формат входных данных

Во входном файле записаны сначала числа N — количество станций ($2 \leq N \leq 100$) и E — количество пар станций, расстояния между которыми заданы ($0 \leq E \leq 10000$). Далее идет E троек чисел, первые два числа каждой тройки задают номера станций (это числа из диапазона от 1 до N), а третье — расстояние между этими станциями (все эти расстояния заданы точно и выражаются вещественными неотрицательными числами не более чем с 3-я знаками после десятичной точки).

Формат выходных данных

В случае, когда восстановить длины перегонов можно однозначно, в выходной файл выведите сначала число 1, а затем $N-1$ вещественное число. Первое из этих чисел должно соответствовать расстоянию от 1-й станции до 2-й, второе — от 2-й до 3-й, и так далее. Все числа должны быть выведены с точностью до 3-х знаков после десятичной точки.

Если приведенная информация о расстояниях между станциями является противоречивой или не позволяет однозначно точно восстановить длины перегонов, выведите в выходной файл одно число 2.

Примеры

a.in	a.out
3 2 1 2 1.250 3 1 3	1 1.250 1.750
4 4 1 2 1.250 1 3 1.255 2 4 0.010 1 1 0.000	1 1.250 0.005 0.005
5 6 1 4 3.000 3 1 2.000 2 4 2.000 1 2 1.000 4 2 2.000 3 5 1.000	1 1.000 1.000 1.000 0.000
3 1 1 1 1	2
3 3 1 2 1.250 1 3 1.300 2 3 1.000	2
3 2 1 2 1.000 1 3 0.005	2
4 2 1 2 1.250 1 4 1.251	2

Задача В Сломанный калькулятор

Имя входного файла: b.in
Имя выходного файла: b.out
Максимальное время работы на одном тесте: 3 секунды
Максимальный объем используемой памяти: 8 мегабайт

У калькулятора есть две ячейки памяти: содержимое первой из них всегда отображается на табло, вторая является буфером. В начальный момент времени на табло калькулятора отображается целое число X , а в буфере записано число 0. У калькулятора работают только две клавиши: «+» и «=». При нажатии на «+» число, которое в данный момент отображено на табло, копируется в буфер. При нажатии на «=» число из буфера прибавляется к числу, отображенному на табло и результат отображается на табло, число в буфере при этом не меняется.

Требуется за наименьшее число нажатий клавиш на калькуляторе добиться того, чтобы на табло было отображено число Y .

Формат входных данных

Входной файл содержит два целых числа X и Y . Каждое из этих чисел по модулю не превышает 10^9 .

Формат выходных данных

В первую строку выходного файла выведите одно число — количество нажатий клавиш, которое потребуется для получения числа Y . Если из числа X получить число Y с помощью указанных операций невозможно, в выходной файл выведите одно число -1 .

Примеры

b.in	b.out	Пояснение (ответ дает следующая последовательность нажатий)
1 1	0	
-2 -6	3	+=
1 8	6	+=+=
2 5	-1	

Задача С Валютные махинации

Имя входного файла: c.in
Имя выходного файла: c.out
Максимальное время работы на одном тесте: 3 секунды
Максимальный объем используемой памяти: 8 мегабайт

Петя, изучая, как меняется курс рубля по отношению к доллару и евро, вывел закон, по которому происходят эти изменения (или думает, что вывел ☺). По этому закону Петя рассчитал, каков будет курс рубля по отношению к доллару и евро в ближайшие N дней.

У Пети есть 100 рублей. В каждый из дней он может обменивать валюты друг на друга по текущему курсу без ограничения количества (при этом курс доллара по отношению к евро соответствует величине, которую можно получить, обменяв доллар на рубли, а потом эти рубли — на евро). Поскольку Петя будет оперировать не с наличной валютой, а со счетом в банке, то он может совершать операции обмена с любым (в том числе и нецелым) количеством единиц любой валюты.

Напишите программу, которая вычисляет, какое наибольшее количество рублей сможет получить Петя к исходу N -го дня.

Законы изменения курсов устроены так, что в течение указанного периода рублевый эквивалент той суммы, которая может оказаться у Пети, не превысит 10^8 рублей.

Формат входных данных

Первая строка входного файла содержит одно число N ($1 \leq N \leq 5000$). В каждой из следующих N строк записано по 2 числа, вычисленных по Петиним законам для соответствующего дня — сколько рублей будет стоить 1 доллар, и сколько рублей будет стоить 1 евро. Все эти значения не меньше 0.01 и не больше 10000. Значения заданы точно и выражаются вещественными числами не более, чем с двумя знаками после десятичной точки.

Формат выходных данных

В выходной файл выведите искомую величину с точностью не менее двух знаков после десятичной точки.

Пример

c.in	c.out
4	4000.00
1 10	
10 5.53	
5.53 1.25	
6 5	

Задача D Двухтуровая олимпиада

Имя входного файла: d.in
Имя выходного файла: d.out
Максимальное время работы на одном тесте: 3 секунды
Максимальный объем используемой памяти: 8 мегабайт

Как известно, личная олимпиада по информатике проходит в два тура. На каждом из туров участники получают какие-то баллы, при этом итоговый балл определяется как сумма полученных баллов. Известны баллы, которые каждый участник получил на каждом из туров. Жюри хочет фальсифицировать итоги олимпиады так, чтобы победил «нужный» участник.

При этом жюри может делать следующие «подтасовки» (можно делать несколько «подтасовок» применительно как к одному и тому же, так и к разным турам):

- Прибавить к результатам всех участников по одному из туров одно и то же положительное число.
- Умножить результаты участников по одному из туров на некоторый коэффициент, больший 1.

При этом должна сохраниться правдоподобность результатов, которая заключается в том, что никто из участников не должен получить больше 100 баллов за каждый из туров.

Определите список участников, которые в результате таких фальсификаций могут оказаться победителями олимпиады (то есть в сумме за два тура иметь не меньше баллов, чем каждый из остальных участников).

Формат входных данных

Во входном файле записано сначала число участников N ($1 \leq N \leq 1000$), затем N пар чисел — результаты каждого участника за 1-й и за 2-й туры (результат участника за тур — это вещественное число от 0 до 100) не более, чем с 3 знаками после десятичной точки.

Формат выходных данных

В выходной файл выведите сначала количество участников, которые смогут стать победителями олимпиады, а затем в возрастающем порядке их номера.

Пример

d.in	d.out
4	2
45 90	2 4
70 80	
0 0	
75 75	

Задача E Черно-белые палиндромы

Имя входного файла: e.in
Имя выходного файла: e.out
Максимальное время работы на одном тесте: 3 секунды
Максимальный объем используемой памяти: 8 мегабайт

Дана полоса клетчатой бумаги длиной N клеток и шириной 1 клетка, в которой некоторые клетки покрашены в черный цвет, а остальные — в белый. Такая полоса называется палиндромом, если последовательность черных и белых клеток при просмотре этой полосы слева направо оказывается такой же, как при просмотре справа налево.

Вам дана полоса длины N . Требуется разрезать ее на полоски, являющиеся палиндромами, так, чтобы количество получившихся полосок было строго меньше величины $\frac{2}{5}N + 3$.

Формат входных данных

Первая строка входного файла содержит число N — длину исходной полосы (N — натуральное число, не превышающее 100000). Далее идет N чисел, описывающих раскраску полосы: 0 означает черную клетку, а 1 — белую.

Формат выходных данных

В выходной файл выведите в возрастающем порядке номера клеток исходной полосы, после которых нужно сделать разрезы.

Примеры

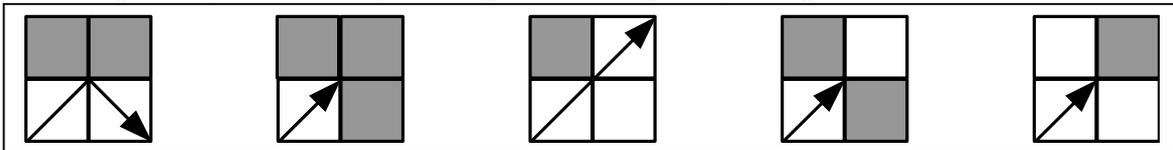
e.in	e.out	Пояснение
6 0 1 0 1 1 0	3 5	Из исходной полосы мы получим 3 полосы-палиндрома, сделав разрезы после 3-й клетки (то есть между 3-й и 4-й) и после 5-й (то есть между 5-й и 6-й)
6 0 1 1 0 0 0	1 3	Данную полосу можно разрезать на 2 полосы-палиндрома, однако по условию не требуется искать решение с минимальным числом получившихся полосок — достаточно, чтобы число полосок удовлетворяло указанному в условии ограничению.
5 0 0 0 0 0		Исходная строка уже является палиндромом, поэтому можно ничего не разрезать

Задача F Луч света в темном царстве

Имя входного файла: f.in
 Имя выходного файла: f.out
 Максимальное время работы на одном тесте: 3 секунды
 Максимальный объем используемой памяти: 8 мегабайт

Темное царство представляет собой лабиринт $N \times M$, некоторые клетки которого окружены зеркальными стенами, а остальные — пустые. Весь лабиринт также окружен зеркальной стеной. В одной из пустых клеток лабиринта поставили светофор, который испускает лучи в 4 направлениях: под 45 градусов относительно стен лабиринта. Требуется изобразить траекторию этих лучей.

Когда луч приходит в угол, через который проходят зеркальные стены, дальше он идет так, как показано на рисунках (серым цветом показаны клетки, которые окружены зеркальными стенами). Аналогичным образом луч ведет себя, когда приходит на границу лабиринта.



Формат входных данных

В первой строке входного файла записаны два натуральных числа N и M — число строк и столбцов в лабиринте (каждое из чисел не меньше 1 и не больше 100). В следующих N строках записано ровно по M символов в каждой — карта лабиринта. Символ * (звездочка) обозначает клетку, окруженную зеркальными стенками, . (точка) — пустую клетку, символ X (заглавная латинская буква X) — клетку, в которой расположен светофор (такая клетка ровно одна).

Формат выходных данных

В выходной файл выведите N строк по M символов в каждой — изображение лабиринта с траекториями лучей. Здесь, как и раньше, * (звездочка) должна обозначать клетки, окруженные зеркальными стенами, . (точка) — пустые клетки, через которые лучи света не проходят, / (слеш) — клетки, через которые луч света проходит из левого нижнего угла в правый верхний (или обратно — из правого верхнего в левый нижний), \ (обратный слеш) — клетки, через которые луч проходит из левого верхнего угла в правый нижний (или обратно), а символ X (заглавная латинская буква X) — клетки, через которые лучи проходят по обеим диагоналям.

Примеры

f.in	f.out
<pre>5 6 ..*...* *X...**</pre>	<pre>./*./\ /..X./ \./X* *X.//* />\X/*.</pre>
<pre>3 3X. ...</pre>	<pre>\./ .X. />\</pre>

Задача G Распаковка строчки

Имя входного файла: g.in
 Имя выходного файла: g.out
 Максимальное время работы на одном тесте: 3 секунды
 Максимальный объем используемой памяти: 8 мегабайт

Будем рассматривать только строчки, состоящие из заглавных латинских букв. Например, рассмотрим строку AAAABCCCCDDDD. Длина этой строки равна 14. Поскольку строка состоит только из латинских букв, повторяющиеся символы могут быть удалены и заменены числами, определяющими количество повторений. Таким образом, данная строка может быть представлена как 4A5C4D. Длина такой строки 7. Описанный метод мы назовем *упаковкой* строки.

Напишите программу, которая берет упакованную строчку и восстанавливает по ней исходную строку.

Формат входных данных

Входной файл содержит одну упакованную строку. В строке могут встречаться только конструкции вида nA, где n — количество повторений символа (целое число от 2 до 99), а A — заглавная латинская буква, либо конструкции вида A, то есть символ без числа, определяющего количество повторений. Максимальная длина строки не превышает 80.

Формат выходных данных

В выходной файл выведите восстановленную строку. При этом строка должна быть разбита на строчки длиной ровно по 40 символов (за исключением последней, которая может содержать меньше 40 символов).

Примеры

g.in	g.out
3A4B7D	AAABBBBDDDDDDDD
22D7AC18FGD	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDAAAAAACFFFFFFFFF FFFFFFFFFGD
95AB	AA AA AAAAAAAAAAAAAAAAAB
40AB39A	AA BAA

Задача Н Дремучий лес

Имя входного файла: h.in
Имя выходного файла: h.out
Максимальное время работы на одном тесте: 3 секунды
Максимальный объем используемой памяти: 8 мегабайт

Просека — это такая прямая линия, которая проходит через лес (то есть деревья есть как с одной стороны от этой линии, так и с другой), и при этом она не проходит ни через одно из деревьев леса, а также не касается деревьев. Будем говорить, что лес является дремучим, если в нем нет ни одной просеки.

На плане леса все деревья изображаются кругами. Никакие два круга не пересекаются и не касаются друг друга. Требуется по этому плану определить, является ли лес дремучим.

Формат входных данных

Во входном файле содержится сначала целое число N — количество деревьев ($1 \leq N \leq 200$). Затем идет N троек чисел, задающих деревья. Первые два числа задают координаты центра, а третье — радиус. Все данные задаются точно, и выражаются вещественными числами, не более чем с 2 знаками после десятичной точки, по модулю не превосходящими 1000.

Формат выходных данных

В первой строке выходного файла должно содержаться сообщение YES, если лес является дремучим, и NO иначе. Во втором случае вторая строка выходного файла должна содержать координаты двух точек, через которые проходит просека. Все координаты нужно выводить с восемью знаками после десятичной точки, координаты не должны превышать 2000, и расстояние между выданными точками должно быть не меньше 100.

Примеры

h.in	h.out
3 0 10 2 5 11 2 12.04 7 2	NO 2.50000000 0.00000000 2.50000000 100.00000000
3 0 0 1 2.05 0 1 1.02 -1.9 1	YES