

Задача А Превосходный делитель

Имя входного файла: divisor.in
 Имя выходного файла: divisor.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мерабайта

Будем говорить, что число a превосходит число b , если сумма цифр a больше суммы цифр числа b , а в случае равенства сумм их цифр, если число a меньше числа b . Например, число 124 превосходит числа 123, так как у первого из них сумма цифр равна семи, а у второго — шести. Также, число 3 превосходит числа 111, так как у них равны суммы цифр, но первое из них меньше.

Дано число n . Найдите такой его делитель (само число n и единица считаются делителями числа n), который превосходит любого другого делителя числа n .

Формат входного файла

Первая строка входного файла содержит целое число n ($1 < n < 105$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

divisor.in	divisor.out
10	5
239	239

Задача В Число e

Имя входного файла: eee.in
 Имя выходного файла: eee.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мерабайта

Выведите в выходной файл округленное до n знаков после десятичной точки число e . Число e , округленное до 25-ти знаков после десятичной точки, равно 2.7182818284590452353602875.

Формат входного файла

Первая строка входного файла содержит целое число n ($0 < n < 25$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

eee.in	eee.out
0	3
25	2.7182818284590452353602875
13	2.7182818284590

Задача С Распаковка строчки

Имя входного файла: unstr.in
 Имя выходного файла: unstr.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мерабайта

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 4AB5C4D. 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 pA , где p — 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

unstr.in	unstr.out
3A4B7D	AAABBBDDDDDD
22D7AC18FGD	DDDDDDDDDDDDDDDDDDDDAAAAACFF FFF FFFF FFF FFFFGD
95AB	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAA
40AB39A	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA BAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Задача D Дремучий лес

Имя входного файла: tree.in
 Имя выходного файла: tree.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мегабайта

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

tree.in	tree.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

Задача E Великая битва

Имя входного файла: battle.in
 Имя выходного файла: battle.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мегабайта

Алеша Попович и Добрыня Никитич сражаются со стаей двух- и трехголовых драконов. Они по очереди взмахивают мечами, и одним махом могут отрубить любое (по своему желанию) число голов, но только у одного дракона. Отрубивший последнюю голову у последнего дракона получает в жены прекрасную принцессу.

Кто из богатырей (начинающий или второй) может получить в жены принцессу независимо от действий другого?

Формат входных данных

Во входном файле записано два числа N и M — количество двух- и трехголовых драконов соответственно (оба числа целые из диапазона от 0 до 100).

Формат выходных данных

В выходной файл выведите сначала число 1 или 2 определяющее, кто из богатырей имеет все шансы получить в жены принцессу (1 — тот, кто начинает, 2 — второй). В случае 1 выведите также все варианты его первого хода, которые к этому приводят: сначала выведите количество различных выигрышных ходов (при этом отрубание одинакового количества голов у разных двухголовых драконов считается одним и тем же ходом, так же и для трехголовых), а затем сами ходы. Каждый ход задается парой чисел: первое число определяет у сколькихголового дракона нужно отрубить головы, а второе — сколько голов нужно отрубить.

Примеры

battle.in	battle.out
2 2	2
1 2	1 2 2 2 3 2

Задача F Коллекционирование марок

Имя входного файла: stamp.in
 Имя выходного файла: stamp.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мегабайта

Вася коллекционирует марки. Для этого у него есть N альбомов вместимостью K_1, K_2, \dots, K_N марок. Вася хочет, чтобы в случае утери одного любого альбома каждая марка осталась у него хотя бы в одном экземпляре. Для этого он покупает каждую марку в двух экземплярах, и наклеивает их в два разных альбома. Какое максимальное количество различных марок при этом может оказаться в его коллекции?

Формат входных данных

Входной файл содержит сначала число N — количество альбомов, а затем N чисел K_1, K_2, \dots, K_N , задающих вместимости альбомов. N — натуральное число из диапазона от 2 до 1000. Вместимость каждого альбома задается натуральным числом, суммарная вместимость всех альбомов не превышает 100000 марок.

Формат выходных данных

В выходной файл выведите сначала число E — максимальное количество различных марок, которое может собрать Вася с соблюдением выдвинутого условия. Затем выведите E пар чисел — каждая пара чисел задает номера двух альбомов, куда будет вклеена очередная марка.

Пример

stamp.in	stamp.out
4 1 2 1 1	2 1 2 2 4

Задача G Перегоны

Имя входного файла: train.in
 Имя выходного файла: train.out
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 Мегабайта

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

train.in	train.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