

Задача А. Лишние пробелы

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

Дана строка. Напишите программу, которая удалит из этой строки все лишние пробелы. Пробел будем считать лишним, если:

- он находится в самом начале строки, до самого первого слова;
- он находится в конце строки, после самого последнего слова;
- несколько пробелов расположены между двумя словами (проще говоря, если слова разделены более чем одним пробелом, тогда все пробелы кроме одного — лишние).

Формат входного файла

Во входном файле записана строка, длина которой не превышает 200 символов. Строка содержит только маленькие латинские буквы и пробелы.

Формат выходного файла

Выведите в выходной файл эту строку без лишних пробелов.

Примеры

spaces.in		spaces.out	
first	test	first	test

Задача В. Сумма двух наоборот

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

Конечно, вы уже сталкивались с такой задачей: даны два числа, требуется вычислить их сумму. Решите ту же задачу, только наоборот. А именно: дана сумма двух целых чисел, найдите два числа, дающие такую сумму.

Формат входного файла

В первой строке входного файла задано одно целое число c .

Формат выходного файла

Выведите в выходной файл два целых числа через пробел — a и b такие, что $a + b = c$. Выведенные числа не должны превышать по модулю 10^9 . Гарантируется, что заданное во входном файле c таково, что ответ при таких ограничениях существует. Если ответов несколько, разрешается выводить любой из них.

Пример

apbtest.in		apbtest.out	
1		3	-2

Задача С. Различные слагаемые

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

В этой задаче требуется найти разбиение натурального числа n на как можно большее количество различных натуральных слагаемых.

Формат входного файла

В первой строке входного файла записано натуральное число n ($1 \leq n \leq 1\,000\,000\,000$).

Формат выходного файла

В первой строке выходного файла выведите k — количество слагаемых. Во второй строке выведите k чисел через пробел — сами слагаемые. Если разбиений с максимальным количеством слагаемых несколько, можно вывести любое из них. Слагаемые можно выводить в любом порядке.

Примеры

sumdiff.in		sumdiff.out	
4		2	1 3
6		3	1 2 3

Задача D. Лишнее число

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

В штаб секретной службы поступило сообщение от одного из агентов. Поступившее сообщение в зашифрованном виде представляет собой последовательность чисел, и лишь специальная программа способна расшифровать его и получить связный текст.

Обычно программа-расшифровщик быстро и бесшумно выдаёт связистам расшифрованный текст, но в этот раз вместо текста от программы поступил сигнал тревоги, свидетельствующий о том, что при пересылке сообщения было взломано или просто повреждено.

Корректное зашифрованное сообщение — это последовательность из $4 \cdot k$ целых чисел, в котором k различных чисел присутствуют по 4 раза каждое; для расшифровки даже не важны значения этих чисел, а важен лишь их порядок.

Однако, изучив зашифрованное сообщение, связисты обнаружили, что в нём $4 \cdot k + 1$ число. При этом ровно одно число является «лишним», то есть при его удалении зашифрованное сообщение становится корректным сообщением из $4 \cdot k$ чисел (возможно, четыре из них равны удалённому числу).

Связисты решили, что на будущее им нужна программа, которая находит такое «лишнее» число автоматически. Помогите им написать такую программу.

Формат входного файла

В первой строке входного файла задано число $N = 4 \cdot k + 1$, где N и k целые, и $1 \leq k \leq 10\,000$. В последующих N строках находятся числа A_1, A_2, \dots, A_N , по одному числу в каждой — зашифрованное сообщение. Известно, что $0 \leq A_i \leq 1\,000\,000$.

Формат выходного файла

В первую строку выходного файла выведите «лишнее» число из набора A_i .

Примеры

excess.in	excess.out
5 4 1 4 4 4	1
9 1 3 3 1 3 3 3 1 1	3

Задача Е. Два числа

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

Даны два целых числа A и B ($1 \leq A, B \leq 100$). Найдите два таких целых числа X и Y , что выполнено равенство $AX + BY = 1$.

Формат входного файла

Во входном файле записаны два числа A и B , разделённые пробелом.

Формат выходного файла

В выходной файл выведите два числа X и Y , разделённые пробелом. Требуется, чтобы выполнялись неравенства $|X| \leq 10\,000$, $|Y| \leq 10\,000$. Если правильных ответов несколько, разрешается вывести любой из них. Если же таких чисел не существует, выведите вместо них два нуля.

Примеры

numbers.in	numbers.out
2 3	2 -1
4 6	0 0
100 51	-5075 9951

Задача F. Точное время

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

У одного холостяка, жившего в глухой сибирской деревне, были настенные и ручные часы. Однажды ручные часы сломались, а настенные остановились, так как хозяин забыл завести их. Радио, телефонов и мобильных в деревне нет. Умный холостяк, поразмыслив, все же придумал, каким образом выставить на настенных часах точное время. Он решил сходить к своему приятелю, у которого, как он знал, были часы, и перед выходом из дома завел свои. Побыв некоторое время у приятеля, он вернулся домой и установил на своих часах точное время. При этом время, которое требовалось, чтобы дойти от своего дома до приятеля холостяк никогда ранее не измерял, но, будучи человеком педантичным, ходил размеренной походкой, поэтому туда и обратно он прошёл за одно и то же время. Какое же время он выставил на часах, придя домой?

Формат входного файла

В первой строке входного файла a — время выхода холостяка из дома, во второй строке b — время прихода к приятелю, в третьей строке c — время ухода от приятеля, в четвертой строке d — время прихода домой от приятеля. Время a и d указано на настенных часах холостяка, а время b и c — на часах приятеля. Каждое время задается в формате: hh:mm (hh — часы, mm — минуты, $00 \leq hh \leq 23$, $00 \leq mm \leq 59$). При этом между временем a и временем d прошло не более суток.

Формат выходного файла

В первую строку выведите точное время, выставленное холостяком на настенных часах по возвращении домой, в формате: hh:mm.

Пример

precise.in	precise.out
14:15 13:05 13:25 14:49	13:32

Задача G. L-Shape

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

Условие вы можете найти по адресу:
<http://acm.math.spbu.ru/trains/101002.pdf>

Задача L.

Задача H. Dales and Hills

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

Условие вы можете найти по адресу:

<http://acm.math.spbu.ru/trains/101002.pdf>

Задача D.

Задача I. Hotel in Ves Lagos

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

Условие вы можете найти по адресу:

<http://acm.math.spbu.ru/trains/101002.pdf>

Задача H.

Задача J. Kids and Prizes

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

Условие вы можете найти по адресу:

<http://acm.math.spbu.ru/trains/101002.pdf>

Задача K.

Задача K. Предок

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

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество вершин в дереве. Во второй строке находится n чисел, i -ое из которых определяет номер непосредственного родителя вершины с номером i . Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число m ($1 \leq m \leq 100\,000$) — количество запросов. Каждая из следующих m строк содержит два различных числа a и b ($1 \leq a, b \leq n$).

Формат выходного файла

Для каждого из m запросов выведите на отдельной строке число 1, если вершина a является одним из предков вершины b , и 0 в противном случае.

Пример

ancestor.in	ancestor.out
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	