

Задача А. Тестирование

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

Одним из способов контроля знаний учащихся является проведение тестирований. На тестировании обычно предлагается несколько вопросов, для каждого из которых необходимо выбрать один из вариантов ответа. Для этого тестируемому выдается специальный бланк с вопросами и вариантами ответов.

Тестирования можно проводить одновременно для достаточно большого количества людей, поэтому возникает вопрос об эффективной обработке заполненных тестируемыми бланков. Во Флатландии эту проблему пытаются решить с помощью применения информационных технологий для автоматической обработки результатов тестирования.

Первым этапом является написание программы, которая бы вычисляла баллы за тест по известной информации о правильных ответах на вопросы и ответах, данных тестируемым. Ваша задача состоит в написании такой программы.

Формат входного файла

Первая строка входного файла содержит число n ($1 \leq n \leq 100000$) вопросов в тесте. Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n — номера правильных вариантов ответов на каждый из вопросов. Третья строка входного файла содержит n целых чисел b_1, b_2, \dots, b_n — номера вариантов, выбранных тестируемым. Для чисел a_i и b_i верны неравенства $1 \leq a_i, b_i \leq 10$.

Формат выходного файла

В выходной файл выведите число вопросов, на которые тестируемый дал правильный ответ.

Примеры

<code>testing.in</code>	<code>testing.out</code>
4 1 2 3 4 1 2 4 3	2
4 1 2 3 4 4 3 2 1	0

Задача В. Торговля акциями

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

В настоящее время на бирже при торговле акциями активно применяются компьютерные системы, которые упрощают и автоматизируют процесс покупки и продажи акций. Некоторые из них даже позволяют вести торговлю вообще без участия человека.

Разумеется, основным критерием, по которому такие системы оцениваются, является прибыль, которую приносит торговля с их применением. Для того, чтобы ее повысить при построении этих систем применяются различные математические методы и модели, такие, как, например, квадратичное программирование, нейронные сети, генетические алгоритмы и т. д.

Основной трудностью при создании таких систем является то, что они должны некоторым образом учитывать изменение стоимости акций в будущем, а также его прогнозировать. Ваша задача несколько проще — курсы продажи и покупки акций за весь период из n дней уже известны, необходимо лишь разработать оптимальную стратегию продаж и покупок. При этом для простоты будем считать, что за эти n дней купить акции можно не более одного раза и продать акции можно также не более одного раза.

Кроме этого, будем считать, что продажа и покупка будет осуществляться только с акциями одного типа. На начало этого периода вы располагаете суммой в x рублей. Для каждого из дней известна цена a_i (от ask — цена предложения), по которой можно купить одну акцию, и цена b_i (от bid — цена спроса), по которой можно одну акцию продать. При этом в соответствии с действующими правилами торгов на бирже разрешается продавать и покупать только целое число акций (например, если у вас есть 5 рублей, а акция стоит 2 рубля, то вы можете купить не более двух акций).

Требуется написать программу, которая по имеющимся данным о стоимости акций в каждый из дней, найдет оптимальную стратегию покупки и продажи акций.

Формат входного файла

Первая строка входного файла содержит два целых числа n и x ($1 \leq n \leq 100000$, $1 \leq x \leq 10^6$). Вторая строка входного файла содержит n целых чисел a_1, \dots, a_n . Третья строка входного файла содержит n целых чисел b_1, \dots, b_n . Для каждого индекса i ($1 \leq i \leq n$) выполняются неравенства $1 \leq b_i \leq a_i \leq 1000$.

Формат выходного файла

В первой строке выходного файла выведите максимальную сумму, которой вы можете обладать по окончании рассматриваемого периода. Во второй строке выведите два числа — номер дня d_1 , в который следует купить акции, и номер дня d_2 , в который эти акции следует продать (должно выполняться неравенство $d_2 > d_1$). При этом подразумевается, что покупается столько акций, сколько их можно купить на x рублей, а потом они все продаются. Если в найденной вами стратегии продавать и покупать акции не требуется, то выведите в выходной файл «-1 -1».

Если существует несколько вариантов оптимальной стратегии, то выведите любой из них.

Примеры

stock.in	stock.out
5 1000 2 3 1 4 3 1 2 1 2 3	3000 3 5
5 1000 10 9 8 7 6 9 8 7 6 5	1000 -1 -1

Задача С. Монотонная последовательность

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

Вася с Петей играют в следующую игру. Вася берет n картонных карточек, и на каждой из них с обеих сторон пишет по числу. После этого он выкладывает карточки в некотором порядке перед Петей. Пронумеруем карточки от 1 до n в том порядке, в котором их выложил Вася.

Петя пытается добиться того, чтобы последовательность чисел, написанных на карточках стала строго монотонной (возрастающей или убывающей). Для этого ему разрешается совершать следующие действия: выбрать два числа l и r , такие что $1 \leq l \leq r \leq n$ и перевернуть каждую из карточек от карточки номер l до карточки номер r .

Напомним, что последовательность c_1, \dots, c_n называется *строго возрастающей*, если выполняются неравенства $c_1 < c_2 < \dots < c_n$, и *строго убывающей*, если выполняются неравенства $c_1 > c_2 > \dots > c_n$.

Напишите программу, которая по описанию карточек определяет, какое минимальное число действий должен совершить Петя для того, чтобы добиться своей цели.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 100000$). Каждая из последующих n строк описывает одну карточку и содержит два числа — a_i написано на ее лицевой стороне, а b_i — на оборотной. Все числа a_i и b_i находятся в диапазоне от 1 до 10^9 .

Формат выходного файла

В первой строке выходного файла выведите минимальное число действий, которое должен совершить Петя. Если Петина цель недостижима, то выведите в выходной файл число -1.

Примеры

monotony.in	monotony.out
3 3 4 1 2 4 1	1
3 1 2 4 1 3 4	-1
6 1 2 3 2 2 3 4 5 6 5 5 6	2

В первом примере для достижения цели Петя может перевернуть карточки со второй по третью. В третьем примере можно, например, первым действием перевернуть карточки со второй по шестую, а вторым — четвертую карточку.

Задача D. Плакат

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

Министерство образования Флатландии недавно приняло решение о начале реализации проекта по использованию социальной рекламы для повышения грамотности населения. Социальная реклама будет размещаться в виде видеороликов на телевидении, плакатов на улицах городов, а также в журналах и газетах.

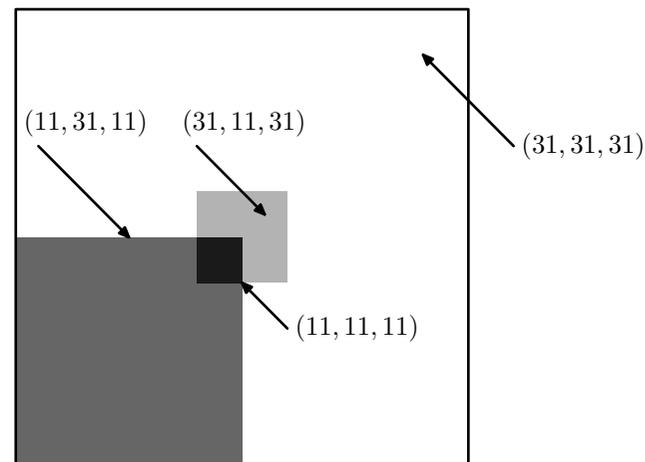
Один из плакатов, подготовленных в рамках этого проекта, выполнен в весьма абстрактном стиле. Плакат имеет форму прямоугольника высотой h и шириной w , который изначально полностью покрашен в белый цвет. На плакате изображены несколько прямоугольников разных цветов. Для того, чтобы напечатать этот плакат службе снабжения министерства образования необходимо выяснить, сколько краски какого цвета для этого требуется.

Так как во Флатландии повсеместно внедрены информационные технологии, то подготовка плаката велась с помощью специальной программы. В этой программе (как и во многих других) цвета представляются в так называемом RGB-формате. Это означает, что цвет кодируется тремя целыми числами, каждое из которых находится в пределах от 0 до 31 (обычно используются значения от 0 до 255, но оборудование, используемое для печати, обладает ограниченными возможностями цветопередачи). Эти числа обозначают интенсивность красной, зеленой и синей компонент цвета (0 соответствует отсутствию соответствующей компоненты, а 31 — ее максимальной интенсивности). Таким образом, черный цвет кодируется как $(0, 0, 0)$, а белый — как $(31, 31, 31)$.

Цвета различных областей плаката определяются следующим образом. Изначально весь плакат заполняется белым цветом (таким образом, исходно все точки имеют цвет $(31, 31, 31)$), после чего выполняется рисование прямоугольников, заданных в описании плаката. При рисовании прямоугольника цветом, который кодируется как (r, g, b) , новые цвета точек, находящихся внутри этого прямоугольника, определяются так: если до рисования точка имела цвет (r_1, g_1, b_1) , то после того, как прямоугольник будет нарисован, ее цвет будет равен $((r_1 + r) \bmod 32, (g_1 + g) \bmod 32, (b_1 + b) \bmod 32)$.

Например, на рисунке приведен плакат, который получается из белого листа размера 10×10 изображением двух прямоугольников: цвета $(12, 0, 12)$ с координатами левого нижнего угла $(0, 0)$ и правого верхнего $(5, 5)$, а затем цвета

$(0, 12, 0)$ с координатами левого нижнего угла $(4, 4)$ и правого верхнего $(6, 6)$.



Таким образом, цвет каждой точки зависит от того, в каких прямоугольниках она лежит, и от того, каковы их цвета. Ваша задача состоит в том, чтобы написать программу, которая по описанию плаката вычислит число различных цветов, которые на нем присутствуют, а также площадь, закрашенную каждым из этих цветов.

Формат входного файла

Первая строка входного файла содержит число n прямоугольников, изображенных на плакате ($0 \leq n \leq 1500$). Вторая строка входного файла содержит два целых числа w и h — соответственно, ширину и высоту плаката ($1 \leq w, h \leq 10^9$). Каждая из последующих n строк описывает один из прямоугольников, изображенных на плакате, и содержит семь целых чисел: $x_1, y_1, x_2, y_2, r, g, b$. Первые два из них — координаты левого нижнего угла прямоугольника, третье и четвертое — координаты правого верхнего угла. Для этих чисел выполняются неравенства $0 \leq x_1 < x_2 \leq w, 0 \leq y_1 < y_2 \leq h$. Пятое, шестое и седьмое числа задают цвет прямоугольника — они равны, соответственно, красной, зеленой и синей компонентам цвета ($0 \leq r, g, b \leq 31$).

Система координат введена таким образом, что левый нижний угол плаката имеет координаты $(0, 0)$, а правый верхний — (w, h) .

Формат выходного файла

В первой строке выходного файла выведите число k различных цветов, которые необходимы для печати плаката. Далее выведите k строк — каждая из них должна

содержать четыре числа — первые три из них должны описывать цвет (первое из них должно быть равно значению красной компоненты, второй — зеленой, а третье — синей), а четвертое должно быть равно площади части плаката, которая будет закрашена этим цветом.

Примеры

poster.in	poster.out
2 10 10 0 0 5 5 12 0 12 4 4 6 6 0 12 0	4 11 11 11 1 11 31 11 24 31 11 31 3 31 31 31 72
0 7 8	1 31 31 31 56
2 10 10 0 0 5 5 20 21 22 4 4 6 6 15 16 17	4 2 4 6 1 14 15 16 3 19 20 21 24 31 31 31 72