

Задача А. Игра в дурака

Имя входного файла: `durak.in`
Имя выходного файла: `durak.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Проказница мартышка, осел, козел и косолапый мишка затеяли сыграть в дурака. Как известно, в первой партии начинает ходить тот, у кого козырь самого маленького достоинства. Поэтому после раздачи карт все четверо одновременно называют достоинство наименьшего козыря, который у них оказался, а именно каждый говорит вслух число от 6 до 14 или 0 (числа больше 10 соответствуют картинкам: валету, даме, королю и тузу, ноль - отсутствию козырей).

Вам известно, какие числа были произнесены. Определите количество гарантированно совравших в этой компании.

В игре используется колода из 36 карт — по 9 карт каждой из 4 мастей. Каждому игроку раздаётся по 6 карт, следующая карта из колоды открывается, и её масть устанавливает козырь для данной игры.

Формат входных данных

Вводятся четыре числа целых числа, которые назвали игроки. Числа разделены пробелами. Каждое из чисел - это либо 0, либо число от 6 до 14.

Формат выходных данных

Выведите единственное число — минимальное количество совравших.

Примеры

<code>durak.in</code>	<code>durak.out</code>
10 7 11 0	0
6 10 10 11	1

Задача В. Контрольная работа

Имя входного файла: `percent.in`
Имя выходного файла: `percent.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Параллель восьмых классов написала контрольную работу. В результате ровно $A\%$ учащихся получили 5, ровно $B\% - 4$, ровно $C\% - 3$, а остальные $D\%$ написали её на 2. Какое минимальное количество школьников должно быть в параллели восьмых классов для того, чтобы могли получиться такие результаты?

Формат входных данных

Вводятся 4 целых числа от 0 до 100 — A, B, C, D ($A + B + C + D = 100$).

Формат выходных данных

Выведите единственное целое положительное число — минимальное возможное количество учащихся в параллели.

Примеры

<code>percent.in</code>	<code>percent.out</code>
40 50 5 5	20

Задача С. Транспортные вопросы

Имя входного файла: `trans.in`
Имя выходного файла: `trans.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

От школы-интерната N . на очный тур Очень Открытой олимпиады прошло N школьников. Для доставки участников на место проведения директор интерната заказывает автобусы и такси. В каждый автобус можно посадить не более 50 школьников, в каждое такси - не более 4 школьников. Почасовая стоимость автобуса составляет A рублей, такси — B рублей (разумеется, $A > B$). На олимпиаду все участники из интерната должны приехать одновременно, то есть в заказанном транспорте должно найтись место сразу для всех.

Помогите директору определить, какое количество автобусов и такси нужно заказать, чтобы потратить как можно меньшую сумму денег на дорогу.

Формат входных данных

Вводятся три целых числа, разделённых пробелами — N , A , B ($1 \leq N \leq 100\,000$, $1 \leq B < A \leq 1\,000$).

Формат выходных данных

Выведите два числа, разделённых пробелами — количество автобусов и количество такси для заказа в оптимальном случае. Если возможных ответов несколько, выведите любой.

Примеры

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

Задача D. Числовые промежутки

Имя входного файла: `numeric.in`
Имя выходного файла: `numeric.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Сегодня в школе Васе рассказывали про числовые промежутки. Каждый из них задаётся парой чисел — своими началом и концом, и информацией о том, включается ли в него каждый из концов. Таким образом, существует четыре типа промежутков:

- **Интервал.** Обозначается (x, y) , включает в себя все числа z : $x < z < y$.
- **Полуинтервалы.** Обозначаются $[x, y)$ и $(x, y]$, включают в себя все такие z , что $x \leq z < y$ и $x < z \leq y$ соответственно.
- **Отрезок.** Обозначается $[x, y]$ и включает в себя все числа z : $x \leq z \leq y$.

В качестве домашней работы Васе досталось посчитать количество целых чисел в каждом из данных промежутков. Поскольку они ещё не проходили вещественных чисел, x и y — рациональные: $x = \frac{a}{b}$, $y = \frac{c}{d}$ (a и c целые, b и d целые положительные).

Рассмотрим пример: $[\frac{3}{2}, 4)$. В данном случае $d = 1$, поэтому вместо $\frac{4}{1}$ пишут просто 4. В этом множестве содержится два целых числа: 2 и 3, а число 4 не содержится.

Помогите Васе с домашней работой — напишите программу, которая по заданному числовому промежутку посчитает количество целых чисел, содержащихся в нём.

Формат входных данных

Первым символом идёт открывающаяся квадратная или круглая скобка. Далее записано число x в формате `a/b` либо `a`, где $|a| \leq 10^9$, $0 < b \leq 10^9$. После следует запятая и пробел. Потом — число y в таком же формате. Далее — закрывающаяся квадратная или круглая скобка. После неё идёт перевод строки и конец файла.

Гарантируется, что данный числовой промежуток не является пустым (то есть содержит в себе хотя бы одно число, не обязательно целое).

Формат выходных данных

По заданному числовому промежутку выведите единственное число — количество целых чисел в нём.

Примеры

<code>numeric.in</code>	<code>numeric.out</code>
<code>[3/2, 4)</code>	2
<code>[-2/4, 5/3]</code>	2
<code>[-1000, 1000]</code>	2001
<code>[-2, 4/3]</code>	4

Задача Е. Делёж яблок

Имя входного файла: `apples.in`
Имя выходного файла: `apples.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Этим летом у бабушки был большой урожай яблок. Она собрала яблоки в корзину и отдала своим K внукам.

Первый внук взял из корзины половину всех яблок и ещё a_1 яблоко (если количество яблок не делилось на два, то результат деления на два он мог округлить как в большую сторону, так и в меньшую). К примеру, если в корзине было 7 яблок и $a_1 = 1$, то он мог взять либо 4, либо 5, а если было 6 яблок и $a_1 = 1$, то он взял ровно 4.

Второй внук взял половину от всех оставшихся яблок и ещё a_2 (если яблок было нечетное количество, то он также мог округлить половину как в большую, так и в меньшую сторону). И так далее, K -ый внук взял половину яблок, оставшихся после $K - 1$ внука, и ещё a_k . В итоге в корзине ничего не осталось.

Теперь они задумались, насколько же большой урожай был у бабушки. Ни один из них не помнит, делилось ли количество яблок на 2 нацело при его выборе, а если нет, то в какую сторону он округлил половину яблок. Внуков интересует минимальное и максимальное изначальное количество яблок в корзине, при которых могли произойти описанные события.

Формат входных данных

Сначала вводится целое положительное число K ($1 \leq K \leq 1000$). Далее записано K целых неотрицательных чисел a_1, \dots, a_K ($0 \leq a_i \leq 1000$).

Формат выходных данных

Выведите два неотрицательных целых числа без ведущих нулей, каждое в новой строке — минимальное и максимальное возможное количество яблок в корзине соответственно.

Примеры

<code>apples.in</code>	<code>apples.out</code>
1	1
1	3
2	1
0 1	7

Система оценки

Тесты состоят из четырёх групп.

- 0) Тесты 1 и 2. Тесты из условия, оцениваются в 0 баллов.
- 1) Тесты 3—7. Тесты с ограничениями $K \leq 2$, $a_i \leq 10$, эти тесты оцениваются в 30 баллов, при этом баллы начисляются только при прохождении всех тестов группы.
- 2) Тесты 8—12. Тесты с ограничениями $K \leq 10$, $a_i \leq 10$, эти тесты оцениваются в 30 баллов, при этом баллы начисляются только при прохождении всех тестов группы.
- 3) Тесты 13—17. Off-line группа, полные ограничения. Каждый тест оценивается в 8 баллов. При этом баллы за тесты этой группы ставятся только тогда, когда программа проходит все тесты групп 1 и 2. Если программа не проходит хотя бы один из тестов групп 1 и 2, то баллы за тесты группы 3 не ставятся.

Задача F. Доставка на дом

Имя входного файла: `delivery.in`
Имя выходного файла: `delivery.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Фирма по грузоперевозкам привезла к воротам загородного дома заказанный домашний кинотеатр в очень большой кубической коробке размерами $1 \times 1 \times 1$ метр. Так как машину на территорию участка не пустили, коробка была сгружена у ворот. Одна из ее сторон (имеется в виду грань куба) помечена как хрупкая — та, рядом с которой расположен экран. Коробка выгружена так, что хрупкая сторона не находится на земле.

Из-за огромных размеров коробки по участку её можно передвигать только перекатывая через ребра. При этом хрупкая сторона не должна оказаться на земле, иначе экран немедленно сломается.

Участок имеет форму прямоугольника размером N на M метров. План участка нарисован на клетчатой бумаге, размер клетки которой соответствует 1 метру. На плане введена система координат так, что левая нижняя клетка плана имеет координату $(1, 1)$, правая нижняя — $(1, M)$, правая верхняя — (N, M) .

Изначально коробка расположена рядом с воротами, в клетке, которая на плане имеет координаты $(1, b)$ (эта клетка расположена у нижней стороны плана участка), а переместить ее надо к двери — на другую клетку с координатами (c, d) . Задано, с какой стороны исходно находится хрупкая сторона. С какой стороны она будет после перекатываний — не важно (важно лишь, чтобы она не оказалась на земле). Участок окружён по периметру забором, поэтому коробку не получится выкатить за пределы участка.

Ваша задача — помочь грузчикам перекатить коробку от ворот до двери дома, не поломав экрана.

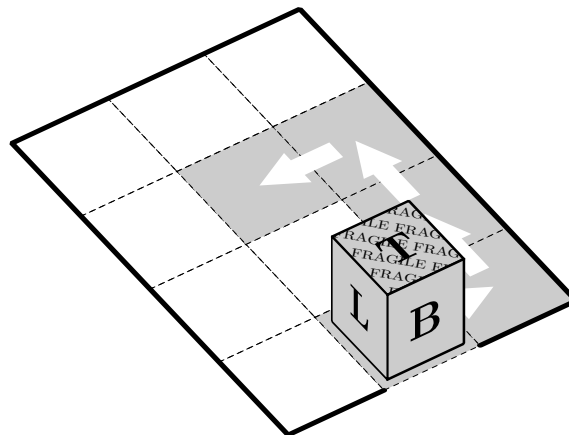


Рис. 1: Рисунок к первому примеру

Формат входных данных

В первой строке вводятся целые числа N, M, b, c, d ($1 \leq N \leq 10\,000, 1 \leq M \leq 10\,000, 1 \leq b \leq M, 1 \leq c \leq N, 1 \leq d \leq M$). Во второй строке содержится одна из букв L, R, T, F, B , описывающая начальное положение хрупкой стороны коробки (слева, справа, сверху, спереди и сзади соответственно). Считается, что задняя сторона коробки повернута в сторону ворот. Ворота и дверь на плане изображаются разными клетками.

Формат выходных данных

Выведите последовательность перекатываний, которая позволит грузчикам выполнить поставленную задачу. Перекатывания обозначаются буквами

- L (перекатывание влево — на единицу уменьшается вторая координата),

- R (перекатывание вправо — на единицу увеличивается вторая координата),
- F (перекатывание вперед — на единицу увеличивается первая координата),
- B (перекатывание назад — на единицу уменьшается первая координата).

Общее количество перекатываний не должно превышать $4(M + N)$ — иначе грузчики не возьмутся за столь тяжелую работу.

Если это невозможно, выведите IMPOSSIBLE.

Примеры

delivery.in	delivery.out
4 3 2 3 2 T	RFFL
2 1 1 2 1 F	IMPOSSIBLE

Система оценки

Тесты состоят из четырёх групп.

- 0) Тесты 1 и 2. Тесты из условия, оцениваются в 0 баллов.
- 1) Тесты 3—15. Одно из чисел N , M равно 1, другое не превосходит 5. Эти тесты оцениваются в 30 баллов, при этом баллы начисляются только при прохождении всех тестов группы.
- 2) Тесты 16—36. В них $N \leq 50$, $M \leq 50$. Эти тесты оцениваются в 40 баллов, при этом баллы начисляются только при прохождении всех тестов группы.
- 3) Тесты 37—39. Off-line группа, полные ограничения. Каждый тест оценивается в 10 баллов. При этом баллы за тесты этой группы ставятся только тогда, когда программа проходит все тесты групп 1 и 2. Если программа не проходит хотя бы один из тестов групп 1 и 2, то баллы за тесты группы 3 не ставятся.

Задача G. Студентам — бесплатно!

Имя входного файла: `painting.in`
Имя выходного файла: `painting.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Зал Большого галактического театра состоит из S рядов, по S мест в каждом ряду. Продажа билетов на каждый спектакль происходит по следующему принципу: первые $S^2 - N$ ценителей прекрасного приобретают билеты на любые места по их вкусу, а оставшиеся N кресел администрация бесплатно выделяет студентам, отдавая дань сложившимся традициям.

Во избежание обвинений в дискриминации по половому признаку, рассаживать студентов по этим N местам необходимо таким образом, чтобы:

- в каждом ряду количество девушек-студенток и количество юношей-студентов различалось бы не более чем на 1;
- на каждой «вертикали мест» (т. е. местах, имеющих один и тот же номер, но расположенных в разных рядах) количество девушек-студенток и количество юношей-студентов также различалось бы не более чем на 1.

Таким образом, после продажи билетов ценителям прекрасного билетёры должны распределить оставшиеся N кресел на женские и мужские с соблюдением этих правил.

Каждое место в зале определяется двумя числами от 1 до S — номером ряда и номером самого места в этом ряду. Студенческое кресло номер i расположено в a_i -м ряду и имеет в нём номер b_i . Поскольку ценители прекрасного могли занять совершенно любые места, числа a_i и b_i могут принимать любые значения от 1 до S . В частности, может оказаться так, что в каком-нибудь ряду не будет ни одного студенческого места.

Ради упрощения работы билетёров администрация обращается к вам с заданием написать программу, которая автоматизирует процесс распределения студенческих мест на мужские и женские.

Формат входных данных

Сначала вводятся два целых числа S и N ($1 \leq S \leq 100\,000$, $1 \leq N \leq \min\{100\,000, S^2\}$). Далее расположены N пар натуральных чисел (a_i, b_i) , не превосходящих S . Гарантируется, что все места различные.

Формат выходных данных

Если искомого способа не существует, выведите `Impossible`. Иначе выведите единственную строку из N символов 'M' (мужское) и 'W' (женское). Символ на i -й позиции соответствует статусу i -го места в той же нумерации, в которой они были перечислены во входных данных.

Примеры

<code>painting.in</code>	<code>painting.out</code>
2 2 2 1 1 2	WW
3 5 1 2 2 3 1 3 2 1 1 1	WMWWM

Система оценки

Тесты состоят из четырёх групп.

- 0) Тесты 1 и 2. Тесты из условия, оцениваются в 0 баллов.
- 1) Тесты 3–19. В них $S \leq 1\,000$, $N \leq 30$. Группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- 2) Тесты 20–30. В них $S \leq 1\,000$, $N \leq 1\,000$. Группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- 3) Тесты 31–34. Off-line группа, полные ограничения. Каждый тест оценивается в 10 баллов (тесты оцениваются независимо друг от друга). При этом баллы за тесты этой группы ставятся только тогда, когда программа проходит все тесты групп 1 и 2. Если программа не проходит хотя бы один из тестов групп 1 и 2, то баллы за тесты группы 3 не ставятся.

Задача Н. Магия числа 23

Имя входного файла: `number23.in`
Имя выходного файла: `number23.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

По заданному числу N найдите натуральное число K , такое что:

- число \overline{KK} (повторённая два раза десятичная запись K) является точным квадратом некоторого натурального числа (см. примеры),
- K при записи в десятичной системе счисления имеет длину от N до $N + 23$ (включительно).

Так, для $N = 1$ условию удовлетворяет, например, число $K = 13223140496$, т.к. оно имеет длину 11, что укладывается в диапазон от 1 до 24, а также число 1322314049613223140496 является точным квадратом натурального числа.

Формат входных данных

Вводится одно натуральное число N ($1 \leq N \leq 2323$).

Формат выходных данных

Выведите искомое число K . Если чисел, удовлетворяющих условию, несколько, выведите любое из них. Если таких чисел не существует, выведите 0.

Примеры

<code>number23.in</code>	<code>number23.out</code>
1	13223140496
11	13223140496
10	29752066116
39	715976331360946745562130177514792899409

Система оценки

Тесты состоят из четырёх групп. В этой задаче нет off-line групп. Баллы за каждую группу начисляются только при прохождении всех тестов этой группы.

- 0) Тесты 1–4, из условия, оцениваются в 0 баллов.
- 1) Тест 5, $N = 13$, оценивается в 30 баллов.
- 2) Тесты 6–14. В них $N \leq 80$. Группа оценивается в 30 баллов.
- 3) Тесты 15–29. Полные ограничения, оценивается в 40 баллов.

Задача I. Как стать призером

Имя входного файла: `roi.in`
Имя выходного файла: `roi.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

К тренеру на занятия по подготовке к олимпиаде ходит N школьников. Для каждого из школьников заданы два параметра: начальный условный опыт A_i и условный интеллект B_i .

Каждое занятие устроено так: тренер подходит к какому-нибудь школьнику и обсуждает с ним возникшие вопросы и проблемы. В результате такого обсуждения условный опыт этого школьника возрастает на B_i (то есть чем выше условный интеллект школьника, тем больше этот школьник может взять из общения с тренером).

За все время подготовки к олимпиаде тренер может подойти ко всем школьникам суммарно не более C раз (он может подходить к разным школьникам, может несколько раз подходить к одному и тому же школьнику). Для того, чтобы школьник стал призером олимпиады, к началу олимпиады его условный опыт должен быть не меньше, чем K .

Напишите программу, которая вычислит максимальное количество призеров олимпиады, которое сможет подготовить тренер.

Формат входных данных

Сначала вводятся натуральные числа N, C, K , задающие количество школьников, количество подходов, которые может сделать учитель, и условный опыт, необходимый, чтобы стать призером олимпиады, соответственно ($1 \leq N \leq 10^6$, $1 \leq C \leq 10^9$, $1 \leq K \leq 10^9$). Далее идет N пар целых неотрицательных чисел A_i, B_i , задающих начальный условный опыт и условный интеллект каждого школьника. Каждое из чисел A_i и B_i не превышает 10^9 .

Формат выходных данных

Выведите одно число — наибольшее количество призеров олимпиады, которое успеет подготовить тренер.

Примеры

<code>roi.in</code>	<code>roi.out</code>
3 5 6 1 1 2 1 4 2	2
4 10 3 0 1 0 1 0 2 2 0	3

Система оценки

Тесты состоят из нулевой группы с тестами из условия, трёх онлайн-групп и двух оффлайн-групп. **Каждая** из групп 1–5 оценивается в 20 баллов, при этом баллы за **каждую** группу начисляются только при прохождении всех тестов этой группы. На 4-й и 5-й группах решение тестируется только если оно проходит все тесты групп 1, 2, 3 (если программа не проходит хотя бы один из тестов групп 1, 2, 3, то баллы за группы 4 и 5 не ставятся).

- 0) Тесты 1 и 2, из условия.
- 1) Тесты 3–10. $N + C \leq 15$, все числа на входе меньше 50.
- 2) Тесты 11–24. $N \leq 1000$, $C \leq 1000$.

- 3) Тесты 25–30. $N \leq 10\,000$.
- 4) Тесты 31–44. $N \leq 100\,000$.
- 5) Тесты 45–54. Полные ограничения.

Задача J. Вася и его друзья

Имя входного файла: `cycles.in`
Имя выходного файла: `cycles.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

В новой игре “Closed Loops 7” игрокам предлагается клетчатая таблица N на M клеток. Ход состоит в том, что очередной игрок рисует *цикл* — замкнутую линию без самопересечений, идущую только по сторонам клеток. Каждый цикл можно нарисовать только один раз за всю игру (при этом, конечно, не запрещается рисовать циклы, пересекающиеся с уже нарисованными). Игроки ходят по очереди. Выигрывает тот, кто рисует последний возможный цикл. К примеру, если $N = 2$, $M = 1$, то циклов всего три и игрок, делающий третий ход, выигрывает:



Вася позвал $K - 1$ друзей поиграть с ним. Чтобы произвести впечатление, он непременно хочет выиграть. Для этого ему нужно узнать, каким по счету игроком он должен быть, чтобы гарантированно одержать победу. Вася наслышан о ваших успехах в программировании, и за помощью он обратился именно к вам.

Формат входных данных

Даны три целых числа: N, M — размеры таблицы ($1 \leq N \leq 100, 1 \leq M \leq 8$) и K — количество игроков ($1 < K \leq 10^9$).

Формат выходных данных

Выведите одно число от 1 до K — каким по счету игроком должен быть Вася, чтобы выиграть.

Примеры

<code>cycles.in</code>	<code>cycles.out</code>
2 1 2	1
1 8 8064	36

Система оценки

Тесты состоят из трёх групп. В этой задаче нет off-line групп.

- 0) Тесты 1 и 2, из условия, оцениваются в 0 баллов.
- 1) Тесты 3–15. $N \cdot M \leq 24$. Эта группа оценивается в 40 баллов, баллы начисляются только при прохождении всех тестов группы.
- 2) Тесты 16–42. Полные ограничения, группа оценивается в 60 баллов. Баллы начисляются только при прохождении всех тестов группы.

Задача К. Отрезки на прямой возвращаются

Имя входного файла: `segments.in`
Имя выходного файла: `segments.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

На прямой задано N попарно различных отрезков $[a_i, b_i]$ ($i = 1, 2, \dots, N, a_i < b_i$). Будем говорить, что отрезок номер i *непосредственно содержится* в отрезке номер j ($i \neq j$), если:

- он полностью принадлежит j -му (то есть $a_j \leq a_i$ и $b_i \leq b_j$),
- среди заданных N отрезков не найдётся такого отрезка (с номером k), что i -й отрезок принадлежит k -му и k -й принадлежит j -му (здесь i, j и k — различные числа).

Ваша задача — для каждого из данных отрезков найти тот, в котором он непосредственно содержится, либо сообщить, что таких нет. Если данный отрезок непосредственно содержится сразу в нескольких — подходит любой из них.

Формат входных данных

Сначала вводится целое число N ($1 \leq N \leq 100\,000$). Далее идут N пар целых чисел a_i, b_i ($-10^9 \leq a_i < b_i \leq 10^9$).

Формат выходных данных

Выведите N чисел. Число номер i должно быть равно номеру отрезка, в котором непосредственно содержится отрезок номер i , либо 0 — если такого не существует.

Если существует несколько решений, выведите любое.

Примеры

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

Система оценки

Тесты состоят из четырёх групп.

- 0) Тест 1, из условия, оценивается в 0 баллов.
- 1) Тесты 2–11. В них $N \leq 100$. Группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- 2) Тесты 12–27. В них $N \leq 10\,000$. Группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- 3) Тесты 28–35. Off-line группа, полные ограничения. Каждый тест оценивается в 5 баллов (тесты оцениваются независимо друг от друга). При этом баллы за тесты этой группы ставятся только тогда, когда программа проходит все тесты групп 1 и 2. Если программа не проходит хотя бы один из тестов групп 1 и 2, то баллы за тесты группы 3 не ставятся.

Задача L. Отрезки на прямой возвращаются—2

Имя входного файла: `segcheck.in`
Имя выходного файла: `segcheck.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

При написании программы, проверяющей ответ участника для предыдущей задачи «Отрезки на прямой возвращаются» (прочитайте её условие!), жюри столкнулось с трудностями, превосходящими сложность самой задачи. С мыслью “почему бы и нет” написание такой программы было решено также включить в комплект задач.

Проверяющей программе доступно три блока информации:

- входные данные в формате, описанном в условии предыдущей задачи,
- ответ некоторого абстрактного участника в формате, также описанном в предыдущем условии,
- ответ жюри.

Ваша задача — написать программу, которая по этим данным определит, правильно ли программа абстрактного участника посчитала ответ.

Формат входных данных

Вход состоит из трёх частей. Первая часть — число N ($1 \leq N \leq 100\,000$) и следом N пар a_i, b_i ($-10^9 \leq a_i < b_i \leq 10^9$). Далее идут N чисел, каждое из которых от 0 до N , i -е равно номеру отрезка, являющегося одним из непосредственно содержащих i -й, либо нулю — по мнению абстрактного участника. Далее идут ещё N чисел в том же формате — ответ жюри на эту задачу.

Входные данные **всегда корректны**. Это означает, например, что ответ участника не нужно проверять на соответствие формату и что ответ жюри точно правильный.

Формат выходных данных

Выведите N строк. В i -й строке должен быть вердикт для i -го отрезка. Выведите `OK`, если ответ абстрактного участника правильный, и `WA` — иначе.

Примеры

segcheck.in	segcheck.out
4	OK
2 3	WA
0 4	WA
1 6	OK
0 5	
2 2 1 0	
3 4 0 0	

Система оценки

Тесты состоят из четырёх групп.

- 0) Тест 1, из условия, оценивается в 0 баллов.
- 1) Тесты 2–11. В них $N \leq 100$. Группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- 2) Тесты 12–27. В них $N \leq 10\,000$. Группа оценивается в 30 баллов, баллы начисляются только при прохождении всех тестов группы.
- 3) Тесты 28–35. Off-line группа, полные ограничения. Каждый тест оценивается в 5 баллов (тесты оцениваются независимо друг от друга). При этом баллы за тесты этой группы ставятся только тогда, когда программа проходит все тесты групп 1 и 2. Если программа не проходит хотя бы один из тестов групп 1 и 2, то баллы за тесты группы 3 не ставятся.