Задача A. Зеленый чай

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	64 мегабайта

Программист Вася любит пить зеленый чай. Для заваривания зеленого чая нельзя использовать кипяток — рекомендуется использовать воду температурой ровно 80 градусов. В комнате, где работает Вася, есть бойлер, который поддерживает постоянную температуру воды t_1 градусов, и кувшин с холодной водой температурой t_2 градусов. У Васи есть кружка, в которой он заваривает чай, и мерная ложка.

Помогите Васе — напишите программу, которая по температурам t_1 и t_2 определит, сколько ложек воды каждого вида (v_1 и v_2) нужно налить в кружку, чтобы получить воду температурой ровно 80 градусов. Если решений несколько, выведите то, которое обладает минимальной суммой $v_1 + v_2$.

Для простоты примем, что в процессе переливания, температура воды остается неизменной, а при смешивании выполняется закон:

$$t_1 \cdot v_1 + t_2 \cdot v_2 = t_3 \cdot (v_1 + v_2),$$

где t_1 и t_2 — температура смешиваемых порций воды, v_1 и v_2 — объемы этих порций, а t_3 — температура получившаяся в результате смешивания.

Формат входных данных

На вход программа получает 2 целых числа, разделенных пробелом: температуры t_1 ($80 \leqslant t_1 \leqslant 100$) и t_2 ($0 \leqslant t_2 < 80$).

Формат выходных данных

Программа должна вывести два целых числа — объемы v_1 и v_2 ($0 \leqslant v_1, v_2 \leqslant 1000$), где v_1 — число ложек воды с температурой t_1, v_2 — число ложек воды с температурой t_2.

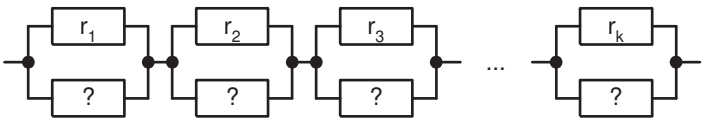
Примеры

стандартный ввод	стандартный вывод
100 20	3 1
100 30	5 2

Задача B. Загадочные резисторы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.4 секунд
Ограничение по памяти:	64 мегабайта

При ремонте электронной платы Вася обнаружил, что плата уже была в ремонте, и один из резисторов на ней заменен на странную конструкцию. Конструкция состояла из k последовательно соединенных звеньев, а каждое звено — из двух параллельно соединенных резисторов.



Причем, в каждом звене, кроме обычного резистора с ясно читаемым номиналом, присутствовал резистор со странной, нестандартной маркировкой. Осмотрев нестандартные резисторы, Вася пришел к выводу, что все они одинаковые, но определить их номинал не смог. Общее сопротивление всей цепи оказалось равным R Ом. Выписав для каждого звена цепи номинал известного резистора, Вася получил ряд целых чисел r_1, r_2, ..., r_k.

Помогите Васе — напишите программу, которая по общему сопротивлению цепи R и известным номиналам r_1, r_2, ..., r_k вычислит номинал загадочных резисторов.

Формат входных данных

Первая строка входных данных содержит два целых числа k ($1 \leqslant k \leqslant 1000$) и R ($1 \leqslant R \leqslant 100000$), разделенных пробелом. Вторая строка содержит k целых чисел, разделенных пробелами: r_1, r_2, ..., r_k ($1 \leqslant r_i \leqslant 100000$; $2R \leqslant r_1 + r_2 + \ldots + r_k$).

Формат выходных данных

Программа должна вывести единственное число — предполагаемый номинал загадочных резисторов. Номинал должен быть выведен с точность до 10^{-6}.

Примеры

стандартный ввод	стандартный вывод
3 11 3 12 30	6.00000000
7 110 15 60 6 45 20 120 70	30.00000000

Замечание

Напомним, что при соединении двух резисторов с номиналами R_1 и R_2 общее сопротивление R вычисляется как $R = R_1 + R_2$ для последовательного соединения и как $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$ для параллельного.

Задача C. Смайлики

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.4 секунд
Ограничение по памяти:	64 мегабайта

Программист Вася пишет новый текстовый редактор. Для того, чтобы отображать почтовую переписку, Вася написал специальный модуль, который извлекает из текстового файла все смайлики и помещает их в отдельную строчку для отображения. К сожалению, в модуль вкралась ошибка, и символы в строчке смайликов перемешались.

Вася знает, что в обрабатываемом письме встречались следующие смайлики: «`;-)`», «`;-(`», «`:)`», «`:(`», «`:-\`», «`:-P`», «`:D`», «`:C`», «`:-0`», «`:-|`», «`8-0`», «`:-E`», «`%0`», «`:-X`», «`:~(`» (символ ~ имеет код 126), «`[:|||:]`». Помогите Васе — напишите программу, которая по строчке символов восстановит смайлики. Если возможно несколько вариантов восстановления, то правильным будет любой из них.

Формат входных данных

На вход подается строка из символов, которые встречаются при записи смайликов, описанных выше. Длина строки не превосходит 10000 символов. Строка не пуста и получена перемешиванием корректной последовательности смайликов.

Формат выходных данных

Программа должна записать восстановленные смайлики, по одному на строку, и завершить запись строкой «LOL» (без кавычек).

Пример

стандартный ввод	стандартный вывод

`|:|;[|)-:]`	`[:|||:]`
	`;-)`
	`LOL`

Задача D. Power play

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Разбирая математическую задачу, программист Вася заметил интересный факт: для чисел 2 и 4 выполняется равенство $2^4 = 4^2$. «Из этого может выйти прекрасная задача для чемпионата» — подумал он. К сожалению, найти другие такие пары чисел Васе не удалось. «Ладно, тогда изменим условия, пусть чисел будет три» — решил Вася. Написанная программа перебора подтвердила, что с тремя числами задача имеет смысл.

Ваша задача: по заданным целым числам a и b найти целое число x, попадающее в диапазон от 1 до 10^{18} (большие числа программа Васи не перебирала), такое что $a^x = x^b$. Если таких чисел несколько, вывести меньшее из них. Если такого числа не существует, нужно вывести 0.

Формат входных данных

Входные данные состоят из двух целых чисел a и b ($2 \leqslant a, b \leqslant 10000$; $a \neq b$), разделенных пробелом.

Формат выходных данных

Целое число x, если решение есть, или 0 (ноль), если решения нет.

Примеры

стандартный ввод	стандартный вывод
2 4	16
2 6	0
2 32	256
100 20	10

Задача E. Печатная плата

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Печатные платы являются одной из основ современной электроники. Для разводки печатных плат существует огромное количество программ, а для перенесения готового рисунка на поверхность платы — не меньшее количество методов (порой весьма экзотических). Одним из способов переноса рисунка является использование графопостроителя — устройства, способного вычерчивать на поверхности непрерывные линии. Напишите программу, которая по рисунку печатной платы составит набор команд для графопостроителя, необходимый для вычерчивания этого рисунка.

Графопостроитель работает на квадратной сетке размером $n \times m$. Квадраты сетки нумеруются слева направо (первая координата) и сверху вниз (вторая координата). Команда для вычерчивания линии состоит из двух чисел — координат начала линии — и последовательности букв «L», «R», «U», «D», которые прочерчивают линию на одну клетку от текущего положения влево, вправо, вверх или вниз соответственно. Например, команда «3 4 DRUL» нарисует квадрат со стороной 1 и левым верхним углом в координатах (3, 4).

С целью минимизации временных затрат на рисование, программа должна построить набор из минимально возможного количества команд (непрерывных линий). Если таких наборов несколько, то решением считается любой из них. Учтите, что:

- Любая клетка платы либо не содержит линии, либо содержит одну линию, либо две пересекающихся (точку пересечения);
- Линия может начинаться и заканчиваться в одной и той же точке;
- Если две линии начинаются в одной точке, то их можно объединить в одну линию;
- Длина линии не может быть меньше 2 (линия не может соединять 2 соседних квадрата);
- Линия или линии не могут начинаться или заканчиваться в соседних клетках (клетки считаются соседними, если имеют общую сторону);
- Линия не может быть проведена поверх уже существующей.

Формат входных данных

Первая строка входных данных содержит целые числа n и m ($1 \leqslant n, m \leqslant 100$) — размер печатной платы. Далее следуют n строк по m символов в каждой. Для кодирования рисунка платы используются следующие символы:

- «.» («Точка», код 0x2E) — пустой квадрат;
- «*» («Звездочка», код 0x2A) — Начало или конец линии, либо точка, где сходятся больше двух линий;
- «-» («Минус», код 0x2D) - Горизонтальная линия;
- «|» («Вертикальная черта», код 0x7C) — Вертикальная линия;
- «L» («Заглавная L», код 0x4C) — Линия, соединяющая верхнюю и правую клетки;
- «J» («Заглавная J», код 0x4A) — Линия, соединяющая верхнюю и левую клетки;
- «r» («Строчная r», код 0x72) — Линия, соединяющая нижнюю и правую клетки;
- «7» («Цифра 7», код 0x37) — Линия, соединяющая нижнюю и левую клетки.

Для любой линии на изображении существует начальная и конечная точки. Линия может начинаться и заканчиваться в одной и той же точке — тогда такая точка обозначается как «*». Через один квадрат проходит либо ровно одна линия, либо, если в этом квадрате происходит пересечение нескольких линий, они считаются соединенными, и квадрат содержит знак «*». Прохождение линий через один квадрат без соединения невозможно. Знаки «*» не могут встречаться в соседних клетках (клетках, имеющих общую сторону).

Формат выходных данных

Первая строка выходных данных содержит число k ($0 \leqslant k \leqslant 10000$) — количество команд. Далее следуют k строк, каждая из которых описывает отдельную команду. Команда начинается с двух целых чисел x и y, разделенных пробелом, — координат начала линии. Далее, также через пробел, следует последовательность из заглавных букв «L», «R», «U» или «D», описывающих прочерчиваемую линию. Буквы идут подряд без пробелов. Команда заканчивается переводом строки.

Примеры

стандартный ввод	стандартный вывод
3 4 r--* *... L--*	1 4 3 LLLUURRR
3 4 r--* L--7 *--J	1 1 3 RRRULLLURRR
3 3 r-* |.| L-J	1 3 1 LLDDRRUU
5 5 ..*.. ..|.. *-*-* ..|.. ..*..	2 3 5 UUUU 1 3 RRRR

Замечание

Обратите внимание, что рисунок платы описывается построчно, а команды используют координаты в формате «столбец строка».

Задача F. Игра в слова

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Алиса и Боб играют в следующую игру:

- перед игроками на бумаге написано слово
- своим ходом игрок может стереть одну букву или две одинаковые буквы, либо вообще все буквы одного вида
- игроки ходят по очереди
- начинает Алиса
- выигрывает игрок, стерший последнюю букву.

Напишите программу, которая определит победителя описанной игры для заданного слова (с учетом того, что игроки играют оптимально).

Формат входных данных

Входные данные содержат единственную строку — начальное слово. Слово состоит из заглавных латинских букв, а его длина не превосходит 40 символов.

Формат выходных данных

Выведите «Alice» (без кавычек), если выигрывает Алиса, или «Bob» (без кавычек), если выигрывает Боб.

Примеры

стандартный ввод	стандартный вывод
ZADACHA	Alice
WORD	Bob

Замечание

Например, для слова «ZADACHA» Алиса первым ходом стирает все буквы «A», остается слово «Z.D.CH.», и выигрывает, так как в последующие ходы игроки вынуждены по очереди стирать по одной букве.

Задача G. Песочные часы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	24 мегабайта

Для измерения времени в Средние века мастеровые использовали песочные часы. Так как такие часы были достаточно редки и дороги, то не у каждого мастера был набор часов, позволяющих отмерить любой интервал времени. Мастера были людьми изобретательными и умудрялись с помощью минимума часов производить очень сложные измерения. Например, с помощью часов на 3 и 5 минут можно отмерить любое количество минут, большее четырех. А сможете ли Вы превзойти наших предков?

У вас есть n (до 4) песочных часов, пронумерованных числами 1, 2, ..., n, песок в которых пересыпается за t_1, t_2, ..., t_n (до 20) минут соответственно. Напишите программу, которая найдет способ отмерить с помощью этих часов k минут. Учтите, что:

- любые часы можно переворачивать только в самом начале измерений, либо в момент, когда в каких-то часах закончился песок
- в такой момент можно перевернуть одновременно любое количество часов — переворачивание часов происходит мгновенно
- часы запрещено класть набок (останавливать время)
- измеряемый интервал в k минут начинается с самого первого переворачивания любых часов
- измеряемый интервал в k минут должен заканчиваться в момент, когда в каких-нибудь часах закончился песок.

Например, пусть у нас есть песочные часы на 3 и 5 минут. Покажем, как с их помощью можно отмерить 7 минут:

- 0 минут. Одновременно переворачиваем часы и на 3 и на 5 минут.
- 3 минуты. Часы на 3 минуты пусты, а в часах на 5 минут есть песок еще на 2 минуты. Переворачиваем часы на 3 минуты.
- 5 минут. Часы на 5 минут пусты, в часах на 3 минуты есть песок еще на 1 минуту сверху, и на 2 минуты снизу. Переворачиваем часы на 3 минуты.
- 7 минут. Часы на 3 минуты пусты. Заданный интервал отмерен.

Формат входных данных

Первая строка входных данных содержит целое число n ($1 \leqslant n \leqslant 4$) — количество песочных часов. Вторая строка содержит целые числа t_1, t_2, ..., t_n ($1 \leqslant t_1 < t_2 < \ldots < t_n \leqslant 20$), разделенные пробелами — номиналы песочных часов. Третья строка содержит целое число k ($1 \leqslant k \leqslant 1440$) — интервал времени, который нужно отмерить.

Формат выходных данных

Первая строка выходных данных содержит -1, если интервал времени k отмерить нельзя. В противном случае, она содержит целое число r — количество моментов времени ($1 \leqslant r \leqslant k$), в которые осуществляется переворачивание часов. Далее следуют r строк, описывающих моменты переворачивания часов.

Каждая такая строка начинается с числа t_i — числа минут от начала отсчета времени до момента данного переворачивания. Далее, через пробел следует число m_i ($0 \leqslant m_i \leqslant n$) — число часов переворачиваемых в момент времени t_i. Далее, через пробел указываются номера переворачиваемых часов.

Строки должны выводиться в порядке увеличения момента времени: $t_i > t_{i-1}$. В момент времени t_i должен заканчиваться песок в хотя бы в одних часах.

Первая строка должна содержать описание начального момента времени ($t_1 = 0$). Последняя строка должна содержать момент времени k и 0 переворачиваемых часов. Строки с временем $t_i > k$ появляться не должны, строки с $m_i = 0$ допускаются и не считаются ошибкой.

Примеры

стандартный ввод	стандартный вывод
2 3 5 7	4 0 2 1 2 3 1 1 5 1 1 7 0
2 3 5 4	-1
2 7 9 13	5 0 2 1 2 7 1 1 9 2 1 2 11 1 2 13 0

Задача H. Мужская разборка

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как-то раз в аудитории 113 развязался большой спор: кто же из двоих студентов пойдет вытирать доску? Проблему начали решать по-мужски! В аудитории лежит коробка с N мелками. Каждый по очереди забирает 1, 5, или 13 мелков. Всё крайне просто: тот, после чьего хода не останется мелков, отправится вытирать исписанную формулами доску. Оба делают свои ходы оптимально.

Кто победит в этом поединке и не будет вытирать злополучную доску?

Формат входных данных

В первой строке вводится число N — количество мелков в аудитории 113 ($1 \leqslant N \leqslant 10000$).

Формат выходных данных

Выведите номер игрока, который выиграет.

Примеры

стандартный ввод	стандартный вывод
4	1
5	2

Задача I. Андрюша и Python

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Это интерактивная задача.

Первое, что должен сделать преподаватель олимпиадного программирования, — это лечить новобранца от серьезной, но вполне излечимой болезни — питонизма.

Марат В. — специалист по лечению данной болезни, но, к сожалению, марганцовка закончилась. Тогда у преподавателей не осталось выбора, кроме как спрятать знания Андрея по злосчастному языку программирования в очень надёжном месте: заначке в паркете аудитории 113.

Паркет в аудитории 113 представляет собой квадрат со стороной $n-1$ на координатной плоскости и содержит n^2 точек с натуральными координатами, а заначка — точка на этой плоскости внутри или на стороне квадрата. Андрей первый год ходит на программирование, поэтому он не знает, где она находится. Единственное, что может делать Андрей — задавать глупые вопросы своему соседу Цукерману. Вопросы бывают трёх видов:

1) Андрей может спросить про конкретную точку, является ли она заначкой;

2) Андрей может спросить про отрезок, заданный координатами его концов, при этом его концы не совпадают — лежит ли загаданная точка на этом отрезке;

3) Андрей может спросить про треугольник, заданный координатами его вершин, при этом треугольник невырожденный — лежит ли загаданная точка внутри или на границе этого треугольника.

В любом запросе координаты точек должны быть натуральными числами.

Цукерман, может отвечать на вопрос либо «Yes», либо «No».

Андрей не хочет, чтобы преподаватели узнали, что он ищет свои знания по языку Python, поэтому он должен задать не более, чем 60 вопросов, чтобы никто ничего не заподозрил.

Помогите Андрюше отгадать загаданную точку в паркете не более чем за 60 вопросов.

Протокол взаимодействия

Сначала на вход вашей программе подается одно целое число n — количество точек на стороне квадрата, в котором нужно искать точку($1 \leqslant n \leqslant 10^8$).

После этого ваша программа может делать запросы:

Для того, чтобы спросить, является ли точка с натуральными координатами (x, y) искомой (запрос типа 1), нужно вывести в выходной поток в отдельной строке «? 1 x y» .

Для того, чтобы спросить, лежит ли искомая точка на отрезке с концами в натуральных точках (x_1, y_1) и (x_2, y_2) (запрос типа 2), нужно вывести в выходной поток в отдельной строке «? 2 x_1 y_1 x_2 y_2» .

Для того, чтобы спросить, лежит ли искомая точка внутри треугольника с вершинами в натуральных точках (x_1, y_1), (x_2, y_2), (x_3, y_3) (запрос типа 3), нужно вывести в выходной поток в отдельной строке «? 3 x_1 y_1 x_2 y_2 x_3 y_3» .

В ответ на запрос во входном потоке будет записана строка «Yes», если ответ положительный, или «No» в противном случае.

В любом запросе координаты точек должны быть натуральными числами и не должны превосходить 10^9.

Если ваша программа сделает более 60 запросов или задаст некорректный запрос (например, запросите вырожденный треугольник), она получит любой вердикт отличный от «Accepted».

Определив, в какой точке плоскости находится заначка со знаниями языка Python, ваша программа должна вывести в выходной поток «! x y», где (x, y) — координаты искомой точки.

Гарантируется, что загаданная точка имеет натуральные координаты, а также загаданная точка фиксированная и не будет меняться в ходе тестирования вашей программы.

Пример

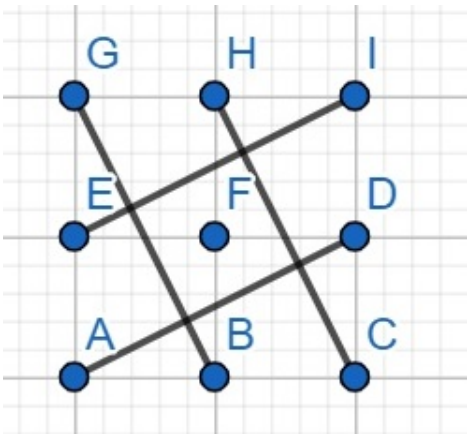
стандартный ввод	стандартный вывод
3	? 2 1 1 3 2
No	? 2 3 1 2 3
No	? 2 3 3 1 2
No	? 2 1 3 2 1
No	? 3 1 1 2 3 3 1
Yes	? 1 2 2
Yes	! 2 2

Замечание

После каждого действия вашей программы выводите символ перевода строки. После вывода очередного запроса обязательно используйте функции очистки потока, чтобы часть вашего вывода не осталась в каком-нибудь буфере. Например, на C++ надо использовать функцию «`fflush(stdout)`», на Java вызов «`System.out.flush()`», на Pascal «`flush(output)`» и «`stdout.flush()`» для языка Python.

ПОЯСНЕНИЕ К ПРИМЕРУ

В тесте из условия была загадана точка с координатами $(2, 2)$ (F на рисунке) на квадрате со стороной 3. Для удобства представлен рисунок.



Первый запрос — принадлежит ли точка отрезку $(1, 1)$ $(3, 2)$ (AD на рисунке)? Ответ — нет.

Второй запрос — принадлежит ли точка отрезку $(3, 1)$ $(2, 3)$ (CH на рисунке)? Ответ — нет.

Третий запрос — принадлежит ли точка отрезку $(3, 3)$ $(1, 2)$ (IE на рисунке)? Ответ — нет.

Четвёртый запрос — принадлежит ли точка отрезку $(1, 3)$ $(2, 1)$ (GB на рисунке)? Ответ — нет.

Пятый запрос — принадлежит ли точка треугольнику $(1, 1)$ $(2, 3)$ $(3, 1)$ (AHC на рисунке)? Ответ — да.

Шестой запрос — является ли точка $(2, 2)$ искомой? Ответ — да.

Выводится ответ ! 2 2.

Задача J. Что-то похожее на проблему Варинга

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Проблема Варинга — теоретико-числовое утверждение, согласно которому для каждого целого $n > 1$ существует такое число $k = k(n)$, что всякое натуральное число N может быть представлено в виде:

$$x_1^n + x_2^n + ... + x_k^n = N$$

с целыми неотрицательными x_1, x_2, ..., x_n.

От вас требуется представить число x в виде $a_1^3 + a_2^3 + ... + a_k^3 = x$, с целыми a_1, ..., a_k и $k \leqslant 5$.

Формат входных данных

В первой строке дано число x ($1 \leqslant x \leqslant 10^{100000}$).

Формат выходных данных

Если такого представления не существует, выведите -1. Иначе, в первой строке выведите число k — количество чисел. Во второй строке выведите k целых чисел a_i ($|a_i| \leqslant 10^{110000}$).

Примеры

стандартный ввод	стандартный вывод
5	5 1 1 1 1 1
17	3 1 2 2

Замечание

В первом примере: $1^3 + 1^3 + 1^3 + 1^3 + 1^3 = 5$

Во втором примере: $1^3 + 2^3 + 2^3 = 17$.

Задача K. Параболическая сортировка

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Профессор Р. — крупнейший специалист в области дискретной математики. Ему уже наскучили все стандартные алгоритмы сортировки, и он придумал новый алгоритм — алгоритм параболической сортировки массива.

Пусть дан массив из N целых различных чисел a_i. Мы хотим его отсортировать таким образом, что первая часть массива строго убывает, а вторая часть массива строго возрастает. При этом мы можем менять местами только соседние элементы. Как первая, так и вторая части массива могут быть длины ноль. Профессора интересует следующий вопрос: какое минимальное количество действий нам понадобится?

Формат входных данных

Во входном потоке $N + 1$ число, первое из которых N ($1 \leqslant N \leqslant 10^5$). Затем следуют числа a_i, по модулю не превосходящие 10^9.

Формат выходных данных

Вывести единственное число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
4 2 3 1 4	1
5 3 2 1 4 5	0