Задача A. Лягушки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Три лягушки сидят на числовой прямой в трёх различных точках с целочисленными координатами a, b, c. Лягушки боятся отступать слишком далеко друг от друга, поэтому прыгать может только одна из крайних лягушек (такая, что слева или справа от нее нет других лягушек) и только в целочисленную точку между двумя другими лягушками, если такая есть. Заметим, что в некоторых положениях ни одна из лягушек прыгнуть не может, назовем их *стабильными*.

Требуется по заданному начальному положению лягушек определить минимальное и максимальное число прыжков, которые могут совершить лягушки, пока не попадут в какое-нибудь стабильное положение.

Формат входных данных

В трёх строках заданы три **различных** целых числа — a, b, c ($1 \leqslant a, b, c \leqslant 10^{18}$), исходные позиции лягушек.

Обратите внимание, что входные данные могут быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип int64 в языке Pascal, тип long long в C и C++, тип long в Java и C#). Язык Python будет корректно работать и с типом int.

Формат выходных данных

Выведите два числа — минимальное и максимальное число прыжков, за которое лягушки могут достичь стабильного состояния.

Примеры

стандартный ввод	стандартный вывод
1 3 4	1 1
1 10 2	2 7
1 2 3	0 0
2 1 5	2 2

Замечание

В первом примере из условия лягушка с позиции 4 может прыгнуть на позицию 2 и образовать стабильное положение $(1, 2, 3)$. Можно показать, что больше одного прыжка они сделать не смогут.

Во втором тесте из условия лягушка с позиции 1 может прыгнуть на позицию 4, а затем лягушка с позиции 10 может прыгнуть на позицию 3, тем самым придя в стабильное положение $(2, 3, 4)$ за два прыжка. Можно показать, что больше 7 прыжков по описанным правилам лягушки сделать не могли.

В третьем тесте из условия лягушки уже находятся в стабильном положении, поэтому прыгать не смогут.

В четвёртом тесте из условия лягушка с позиции 1 может прыгнуть на позицию 4, а затем лягушка с позиции 5 может прыгнуть на позицию 3, тем самым придя в положение $(2, 3, 4)$ за два прыжка. Можно показать, что больше двух прыжков лягушки сделать не могли.

Система оценки

В данной задаче 25 тестов, помимо тестов из условия, каждый из них оценивается в 4 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant a, b, c \leqslant 30$, наберут не менее 40 баллов.

Решения, корректно работающие при $1 \leqslant a, b, c \leqslant 2000$, наберут не менее 60 баллов.

Задача B. Автомобиль «Пантера»

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	512 мегабайт

Однажды утром Глеб с ужасом осознал, что проспал, а пары в «Высшем университете» начинаются уже скоро. Опоздать было бы не так страшно, если бы он их и не вёл. К счастью, автомобиль Глеба «Пантера» довольно мощный: для упрощения будем считать, что за одну секунду он может сначала или увеличить скорость на 1, или уменьшить скорость на 1, или не менять её, а после этого его автомобиль проезжает x метров, где x — его текущая скорость в метрах в секунду. Потом он снова принимает решение об изменении скорости. Начальная скорость автомобиля преподавателя в момент, когда он только выезжает из дома, равна нулю. Путь до университета от его дома не близкий: нужно проехать d метров.

Глеб, обеспокоенный за знания своих студентов, хочет попасть в университет как можно раньше, чтобы успеть подготовиться к проведению лекции. К сожалению, у проезда на парковку университета установлен датчик движения: если скорость проезжающего транспортного средства будет превышать s, то администрация университета выпишет нарушителю дисциплинарное взыскание. Конечно, Глеб не хочет его получить, поэтому при въезде в университет, когда автомобиль проехал все d метров его скорость x должна быть не больше s.

Пока Глеб собирался на работу, его заинтересовал вопрос, а за какое минимальное время он может доехать до работы, если будет действовать оптимально, но не нарушать правила при въезде в университет. Так как преподаватель будет занят скоростным вождением, ответить на этот вопрос честь выпала вам!

Формат входных данных

В двух строках заданы два целых числа — d, s ($1 \leqslant d \leqslant 10^{18}$, $0 \leqslant s \leqslant 10^{18}$), расстояние до университета и ограничение на конечную скорость.

Обратите внимание, что входные данные могут быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип int64 в языке Pascal, тип long long в C и C++, тип long в Java и C#). Язык Python будет корректно работать и с типом int.

Формат выходных данных

Выведите единственное число — минимальное время, за которое Глеб может доехать до университета, не получив дисциплинарное взыскание.

Примеры

стандартный ввод	стандартный вывод
2 1	2
10 0	7

Замечание

В первом тесте из условия Глебу выгодно действовать следующим образом:

1. Увеличить скорость на один, проехать один метр; скорость 1 м/с, проехал 1 метр.
2. Не менять скорость, проехать один метр; скорость 1 м/с, проехал 2 метра.

Таким образом, он потратит 2 секунды, и его конечная скорость будет равно 1 м/с, что не превышает s.

Во втором тесте из условия Глебу выгодно действовать, например, так:

1. Увеличить скорость на один, проехать один метр; скорость 1 м/с, проехал 1 метр.
2. Увеличить скорость на один, проехать два метра; скорость 2 м/с, проехал 3 метра.
3. Увеличить скорость на один, проехать три метра; скорость 3 м/с, проехал 6 метров.
4. Уменьшить скорость на один, проехать два метра; скорость 2 м/с, проехал 8 метров.
5. Уменьшить скорость на один, проехать один метр; скорость 1 м/с, проехал 9 метров.
6. Не менять скорость, проехать один метр; скорость 1 м/с, проехал 10 метров.
7. Уменьшить скорость на один; скорость 0 м/с, проехал 10 метров.

Таким образом, он потратит 7 секунд, и его конечная скорость будет равно 0 м/с, что не превышает s.

Система оценки

В данной задаче 25 тестов, помимо тестов из условия, каждый из них оценивается в 4 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $s = 0$, наберут не менее 20 баллов.

Решения, корректно работающие при $1 \leqslant d \leqslant 10^4$, наберут не менее 40 баллов.

Решения, корректно работающие при $1 \leqslant d \leqslant 10^9$, наберут не менее 60 баллов.

Задача BZ. Автомобиль «Пантера»

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	512 мегабайт

Однажды утром Глеб с ужасом осознал, что проспал, а пары в «Высшем университете» начинаются уже скоро. Опоздать было бы не так страшно, если бы он их и не вёл. К счастью, автомобиль Глеба «Пантера» довольно мощный: для упрощения будем считать, что за одну секунду он может сначала или увеличить скорость на 1, или уменьшить скорость на 1, или не менять её, а после этого его автомобиль проезжает x метров, где x — его текущая скорость в метрах в секунду. Потом он снова принимает решение об изменении скорости. Начальная скорость автомобиля преподавателя в момент, когда он только выезжает из дома, равна нулю. Путь до университета от его дома не близкий: нужно проехать d метров.

Глеб, обеспокоенный за знания своих студентов, хочет попасть в университет как можно раньше, чтобы успеть подготовиться к проведению лекции. К сожалению, на проезде на парковку университета установлен шлагбаум, поэтому скорость автомобиля после того, как он проехал ровно d метров должна быть равна 0.

Пока Глеб собирался на работу, его заинтересовал вопрос, а за какое минимальное время он может доехать до работы, если будет действовать оптимально. Так как преподаватель будет занят скоростным вождением, ответить на этот вопрос честь выпала вам!

Формат входных данных

Вводится одно целое число — d ($1 \leqslant d \leqslant 10^{18}$), расстояние до университета.

Обратите внимание, что входные данные могут быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип int64 в языке Pascal, тип long long в C и C++, тип long в Java и C#). Язык Python будет корректно работать и с типом int.

Формат выходных данных

Выведите единственное целое число — исходное минимальное время, за которое Глеб может доехать до университета, в точности остановившись перед шлагбаумом.

Примеры

стандартный ввод	стандартный вывод
2	3
10	7

Замечание

В первом тесте из условия Глебу выгодно действовать следующим образом:

1. Увеличить скорость на один, проехать один метр; скорость 1 м/с, проехал 1 метр.
2. Не менять скорость, проехать один метр; скорость 1 м/с, проехал 2 метра от дома.
3. Уменьшить скорость на один; скорость 0 м/с, проехал 2 метра от дома.

Таким образом, он потратит 3 секунды, и его конечная скорость будет равно 0 м/с. Во втором тесте из условия Глебу выгодно действовать, например, так:

1. Увеличить скорость на один, проехать один метр; скорость 1 м/с, проехал 1 метр.
2. Увеличить скорость на один, проехать два метра; скорость 2 м/с, проехал 3 метра.
3. Увеличить скорость на один, проехать три метра; скорость 3 м/с, проехал 6 метров.

4. Уменьшить скорость на один, проехать два метра; скорость 2 м/с, проехал 8 метров.
5. Уменьшить скорость на один, проехать один метр; скорость 1 м/с, проехал 9 метров.
6. Не менять скорость, проехать один метр; скорость 1 м/с, проехал 10 метров.
7. Уменьшить скорость на один; скорость 0 м/с, проехал 10 метров.

Таким образом, он потратит 7 секунд, и его конечная скорость будет равно 0 м/с.

Система оценки

В данной задаче 25 тестов, помимо тестов из условия, каждый из них оценивается в 4 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant d \leqslant 10^2$, наберут не менее 20 баллов

Решения, корректно работающие при $1 \leqslant d \leqslant 10^4$, наберут не менее 40 баллов.

Решения, корректно работающие при $1 \leqslant d \leqslant 10^9$, наберут не менее 60 баллов.

Решения, корректно работающие при $1 \leqslant d \leqslant 10^{12}$, наберут не менее 80 баллов.

Задача C. 2022

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	512 мегабайт

Эвелине на Новый год подарили массив a из n неотрицательных целых чисел, каждое из которых не превосходит 2022. Её заинтересовал вопрос, сколько в этом массиве существует различных четверок индексов (индексы внутри одной четверки могут совпадать) таких, что сумма соответствующих элементов массива равна 2022. Формально, она хочет понять, сколько существует четверок $1 \leqslant i, j, k, h \leqslant n$, для которых выполняется $a_i + a_j + a_k + a_h = 2022$.

Уже наступил февраль, а Эвелина все еще не успела посчитать ответ на вопрос, потому что массив слишком большой. Но она смогла запомнить его и рассказала о своем массиве вам, чтобы получить помощь с поиском ответа.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество элементов массива. Во второй строке заданы n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 2022$) — элементы массива Эвелины.

Формат выходных данных

Так как ответ может быть слишком большим, вам необходимо вывести остаток от деления количества подходящих четверок на 1000000007 ($10^9 + 7$).

Примеры

стандартный ввод	стандартный вывод
4 1 1 1 1	0
2 500 511	6
4 129 45 1000 848	24

Замечание

В первом примере не существует четверок с суммой 2022.

Во втором подходят четверки $(1, 1, 2, 2)$, $(1, 2, 1, 2)$, $(1, 2, 2, 1)$, $(2, 1, 1, 2)$, $(2, 1, 2, 1)$, $(2, 2, 1, 1)$.

В третьем примере подходят все 24 четверки попарно различных индексов. Например, $(1, 2, 3, 4)$ или $(4, 1, 3, 2)$.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant n \leqslant 30$, наберут не менее 10 баллов.

Решения, корректно работающие при $1 \leqslant n \leqslant 150$, наберут не менее 30 баллов.

Решения, корректно работающие при $1 \leqslant n \leqslant 1500$, наберут не менее 50 баллов.

Задача CZ. 2022

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Эвелине на Новый год подарили массив a из n неотрицательных целых чисел, каждое из которых не превосходит 2022. Её заинтересовал вопрос, сколько в этом массиве существует различных пар индексов, у которых первый индекс в паре меньше второго, таких, что сумма соответствующих элементов массива равна 2022. Формально, она хочет понять, сколько существует пар $1 \leqslant i < j \leqslant n$, для которых выполняется $a_i + a_j = 2022$.

Уже наступил февраль, а Эвелина все еще не успела посчитать ответ на вопрос, потому что массив слишком большой. Но она смогла запомнить его и рассказала о своем массиве вам, чтобы получить помощь с поиском ответа.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество элементов массива.

Во второй строке заданы n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 2022$) — элементы массива Эвелины.

Формат выходных данных

Выведите одно число — количество подходящих пар.

Примеры

стандартный ввод	стандартный вывод
2 1 2022	0
4 1000 1022 1001 1021	2
5 700 1 1 2021 2021	4

Замечание

В первом примере не существует пар с суммой 2022.

Во втором примере подходят пары $(1, 2)$, $(3, 4)$.

В третьем примере подходят пары $(2, 4)$, $(2, 5)$, $(3, 4)$, $(3,\ 5)$.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant n \leqslant 1500$, наберут не менее 50 баллов.

Задача D. Максимальный XOR

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Два числа a и b записаны в шестнадцатеричной системе счисления. Запись обоих имеет длину n. Вы можете сколько угодно раз менять две соседние цифры местами в любом из чисел. Какое максимальное значение может быть у результата применения побитовой операции XOR к получившимся после применения таких перестановок числам?

Эта операция определена над двоичным представлением чисел.

Определим операцию побитового исключающего «ИЛИ» (XOR). Пусть даны два целых неотрицательных двоичных числа x и y длины k (возможно с ведущими нулями): $x_{k-1}\dots x_2x_1x_0$ и $y_{k-1}\dots y_2y_1y_0$. Здесь x_i это i-й бит числа x, а y_i это i-й бит числа y. Пусть $r = x$ XOR y — результат операции XOR над числами x и y. Тогда двоичной записью r будет $r_{k-1}\dots r_2r_1r_0$, где:

$$r_i = \begin{cases} 1, \text{ если } x_i \neq y_i \\ 0, \text{ если } x_i = y_i \end{cases}$$

Формат входных данных

В первой строке содержится одно целое число n ($1 \leqslant n \leqslant 100\,000$) — длина записи чисел. Во второй строке задана запись числа a. В третьей строке задана запись числа b.

Буквы A, B, C, D, E, F отвечают за цифры 10, 11, 12, 13, 14, 15 в шестнадцатеричной системе счисления соответсвенно. Записи могут содержать ведущие нули.

Формат выходных данных

В единственной строке вам необходимо вывести одно шестнадцатеричное число длины n, которое является ответом на вопрос из условия.

Примеры

стандартный ввод	стандартный вывод
2 0F 0E	FE
3 000 000	000
6 010110 011000	111110

Замечание

В первом примере можно поменять две соседние цифры в первом числе, получится F0 XOR 0E = FE.

Во втором примере любая перестановка цифр не меняет a XOR b. Обратите внимание, что длина выводимого числа должна быть равна n, поэтому надо выводить лидирующие нули.

В третьем примере можно получить 101010 из a и 010100 из b.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant n \leqslant 8$, наберут не менее 10 баллов.

Решения, корректно работающие при a, состоящем только из цифры 0, наберут не менее 10 баллов.

Решения, корректно работающие, когда a и b состоят только из цифр 0 и 1, наберут не менее 10 баллов.

Решения, корректно работающие, когда a состоит только из цифр 0, 1, 2, 3, а b состоит только из цифр 0, 4, 8, C, наберут не менее 10 баллов.

Решения, корректно работающие, когда a состоит из не более двух различных цифр и b состоит из не более двух различных цифр, наберут не менее 30 баллов.

Задача DZ. Максимальный XOR

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Два числа a и b записаны в двоичной системе счисления. Запись обоих имеет длину $2n$. Обе записи разбиты на n блоков по 2 стоящие рядом цифры. В каждом из чисел вы можете сколько угодно раз менять два произвольных блока местами. Какое максимальное значение может быть у результата применения побитовой операции XOR к получившимся числам?

Определим операцию побитового исключающего «ИЛИ» (XOR). Пусть даны два целых неотрицательных двоичных числа x и y длины k (возможно с ведущими нулями): $x_{k-1}\ldots x_2x_1x_0$ и $y_{k-1}\ldots y_2y_1y_0$. Здесь x_i это i-й бит числа x, а y_i это i-й бит числа y. Пусть $r = x$ XOR y — результат операции XOR над числами x и y. Тогда двоичной записью r будет $r_{k-1}\ldots r_2r_1r_0$, где:

$$r_i = \begin{cases} 1, \text{ если } x_i \neq y_i \\ 0, \text{ если } x_i = y_i \end{cases}$$

Формат входных данных

В первой строке содержится одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество блоков по две цифры в записях обоих числах. Во второй строке задана запись числа a. В третьей строке задана запись числа b.

Для удобства блоки разделены символом «|».

Формат выходных данных

В единственной строке вам необходимо вывести одно двоичное число из n блоков, которое является ответом на задачу, в таком же блочном формате, в каком заданы числа a и b.

Примеры

стандартный ввод	стандартный вывод
2 00|11 00|10	11|10
3 00|00|00 00|00|00	00|00|00
3 10|10|01 00|01|01	11|11|01

Замечание

В первом примере можно поменять два соседних блока в первом числе, получится $11|00$ XOR $00|10 = 11|10$.

Во втором примере любая перестановка цифр не меняет a XOR b. Обратите внимание, что длина выводимого числа должна состоять из n блоков, поэтому надо выводить блоки из лидирующих нулей.

В третьем примере можно получить 101001 из a и 010100 из b.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant n \leqslant 8$, наберут не менее 20 баллов.

Решения, корректно работающие при a, состоящем только из цифры 0, наберут не менее 10 баллов.

Решения, корректно работающие, когда a и b состоят только из блоков 00 и 01, наберут не менее 10 баллов.

Решения, корректно работающие, когда a состоит только из блоков 00 и 01, а b состоит только из блоков 00 и 10, наберут не менее 20 баллов.

Задача E. Массив сумм

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Юля выписала на доску n последовательных натуральных чисел $a, a+1, \ldots, a+n-1$ и написала под каждым из них сумму его цифр в десятичной записи, под i-м числом было выписано sum_i.

После этого Юра стёр исходные числа и оставил только их суммы цифр. От вас требуется восстановить первое число в исходной последовательности a.

Формат входных данных

В первой строке содержится одно целое число n ($2 \leqslant n \leqslant 100\,000$) — длина исходной последовательности.

В следующей строке содержатся n целых чисел $sum_1, sum_2, \ldots, sum_n$ ($1 \leqslant sum_i \leqslant 90$) — суммы цифр чисел исходной последовательности.

Гарантируется, что для всех тестов существует подходящее a, такое что $1 \leqslant a \leqslant 10^{18}$.

Формат выходных данных

Выведите одно число a ($1 \leqslant a \leqslant 10^{18}$) — первое число исходной последовательности. В случае, если существует несколько подходящих a, можно вывести любое.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	1
5 14 15 16 8 9	77

Замечание

В первом тестовом примере сумма цифр 1 равняется 1, сумма цифр 2 равняется 2, сумма цифр 3 равняется 3, что соотносится с массивом sum, поэтому $a = 1$ подходит под условие задачи.

Во втором тестовом примере сумма цифр 77 равняется 14, сумма цифр 78 равняется 15, сумма цифр 79 равняется 16, сумма цифр 80 равняется 8, сумма цифр 81 равняется 9, что соотносится с массивом sum, поэтому $a = 77$ подходит под условие задачи.

Система оценки

В данной задаче 25 тестов, помимо тестов из условия, каждый из них оценивается в 4 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $1 \leqslant n \leqslant 2$, наберут не менее 20 баллов.

Решения, корректно работающие при $1 \leqslant n \leqslant 100$, наберут не менее 40 баллов.

Решения, корректно работающие при $1 \leqslant n \leqslant 1000$, наберут не менее 64 баллов.

Решения, корректно работающие при $1 \leqslant n \leqslant 10\,000$, наберут не менее 80 баллов.

Задача F. Возрастающие пути

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	512 мегабайт

Дано дерево на n вершинах (связный неориентированный ациклический граф с $n-1$ рёбрами), где у каждого ребра есть вес w.

Назовём простой путь длины k *возрастающим*, если существует такое целое $x \geqslant 2$, что вес первого ребра пути делится на x, второго ребра — делится на x^2, $\ldots$, вес k-го ребра делится на x^k.

Требуется найти максимальную длину k *возрастающего* пути, где k — количество рёбер в нём.

Формат входных данных

В первой строке вводится единственное целое число n ($1 \leqslant n \leqslant 100\,000$) — число вершин в дереве.

В следующих $n-1$ строках вводятся по три целых числа u, v, w ($1 \leqslant u \leqslant n$, $1 \leqslant v \leqslant n$, $u \neq v$, $1 \leqslant w \leqslant 10^7$) — номера вершин, которые соединяет очередное ребро, и его вес.

Формат выходных данных

Выведите одно целое число k — максимальную длину возрастающего пути.

Примеры

стандартный ввод	стандартный вывод
4 1 2 8 1 3 6 1 4 3	2
6 1 2 2 2 3 4 3 4 2 4 5 4 5 6 8	3

Замечание

Простым путем называется такой путь, что все вершины в нем различны.

В 1-м примере есть путь длины 2: 3 — 1 — 2. Тогда для него подходящий $x = 2$. Можно показать, что возрастающего пути большей длины не существует.

Во 2-м примере есть путь длины 3: 3 — 4 — 5 — 6. Тогда для него подходящий $x = 2$. Можно показать, что возрастающего пути большей длины не существует.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Ограничения		Необходимые группы
		n	Дополнительно	
0	0		Тесты из условия	
1	28	$n \leqslant 1000$		0
2	12	$n \leqslant 100\,000$	Степени вершин не больше двух	
3	20	$n \leqslant 100\,000$	Веса рёбер — степени двойки	
4	11	$n \leqslant 100\,000$	Вес каждого ребра не больше 10000	0
5	29	$n \leqslant 100\,000$	**Offline-проверка**	0–4

Задача G. Башни 2.0

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В компьютерной игре есть n башен, высота i-й башни равна a_i метров. Определим *расстояние* между двумя башнями с индексами i и j как $|i-j|$. Разрешается прыгнуть с i-й башни на j-ю башню тогда и только тогда, когда не существует такого индекса $1 \leqslant k \leqslant n$, такого, что расстояние от i-й до j-й башни не меньше расстояния от i-й башни до k-й башни, и k-я башня имеет большую высоту, чем j-я. Башня j *достижима* из башни i если существует последовательность корректных прыжков, которая начинается в i-й башне и заканчивается в j-й. Посчитайте для каждой башни количество достижимых из неё башен, включая её саму.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \leqslant n \leqslant 500\,000$) — количество башен.

Вторая строка входных данных содержит n чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_n \leqslant 10^9$) — высоты башен.

Формат выходных данных

Выведите n чисел, i-е из которых должно быть равным количеству башен, достижимых из i-й башни.

Примеры

стандартный ввод	стандартный вывод
5 7 6 3 4 10	2 3 4 2 1
7 1 1 1 2 2 1 1	5 5 3 2 2 3 4

Замечание

В первом примере с 1-й башни можно прыгнуть на башни 1 и 5. Любая другая башня имеет меньшую высоту, чем башня 1, поэтому туда нельзя прыгнуть (в качестве k можно выбрать 1). Множество достижимых из 1-й башни также состоит из башен 1 и 5. Со второй башни можно прыгнуть на башни 1, 2, и 5, они же являются множеством достижимых. С третьей башни можно прыгнуть на башни 2, 3, 5. Однако, башня 1 также является достижимой, поскольку можно сделать два прыжка: $3 \to 2 \to 1$. Таким образом, получается 4 достижимые башни. С 4-й башни можно прыгнуть на башни 4 и 5, они же являются единственными достижимыми. Из 5-й башни достижима только она сама.

Во втором примере из 1-й и из 2-й башни достижимы башни $1, 2, 3, 4, 5$. Из 3-й башни достижимы башни $3, 4, 5$. Из 4-й и 5-й башни достижимы башни $4, 5$. Из 6-й башни достижимы башни $4, 5, 6$. Из 7-й башни достижимы башни $4, 5, 6, 7$.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Обозначим за C высоту самой высокой башни.

Группа	Баллы	Ограничения			Необходимые группы
		n	C	Дополнительно	
0	0			Тесты из условия	
1	18	$n \leqslant 100$	$C \leqslant 10^9$		0
2	11	$n \leqslant 2000$	$C \leqslant 10^9$	Все a_i различны	
3	9	$n \leqslant 200\,000$	$C \leqslant 2$		
4	16	$n \leqslant 200\,000$	$C \leqslant 3$		3
5	19	$n \leqslant 10\,000$	$C \leqslant 10^9$		0–2
6	27	$n \leqslant 500\,000$	$C \leqslant 10^9$	**Offline-проверка**	0–5

Задача GD. Башни 3.0

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

В компьютерной игре есть n башен, высота i-й башни равна a_i метров. Определим *расстояние* между двумя башнями с индексами i и j как $|i - j|$. Разрешается прыгнуть с i-й башни на j-ю тогда и только тогда, когда не существует такого индекса $1 \leqslant k \leqslant n$, такого, что расстояние от i-й до j-й башни не меньше расстояния от i-й башни до k-й, и k-я башня имеет большую высоту, чем j-я. Башня j *достижима* из башни i если существует последовательность корректных прыжков, которая начинается в i-й башне и заканчивается в j-й.

Вам даны q запросов вида (u, v, l, r). Для каждого запроса посчитайте количество индексов $l \leqslant k \leqslant r$, таких, что k-я башня достижима из u-й башни и из v-й башни. **Обратите внимание, что во многих подзадачах выполняется ограничение $u = v$, $l = 1$, $r = n$, то есть ответом на запрос будет общее число башен, достижимых из u.**

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \leqslant n \leqslant 500\,000$) — количество башен.

Вторая строка входных данных содержит n чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_n \leqslant 10^9$) — высоты башен.

Третья строка входных данных содержит одно целое число q ($1 \leqslant q \leqslant 500\,000$) — количество запросов.

Следующие q строк описывают запросы. i-я из них описывает i-й запрос и содержит четыре целых числа u_i, v_i, l_i, r_i ($1 \leqslant u_i, v_i, \leqslant n$, $1 \leqslant l_i \leqslant r_i \leqslant n$) — индексы вершин запроса и границы отрезка запроса.

Формат выходных данных

Выведите n чисел, i-е из которых должно быть равным ответу на i-й запрос.

Примеры

стандартный ввод	стандартный вывод
5 7 6 3 4 10 5 1 1 1 5 2 2 1 5 3 3 1 5 4 4 1 5 5 5 1 5	2 3 4 2 1
7 1 1 1 2 2 1 1 7 1 1 1 7 2 2 1 7 3 3 1 7 4 4 1 7 5 5 1 7 6 6 1 7 7 7 1 7	5 5 3 2 2 3 4
7 6 8 9 3 5 10 1 5 1 3 2 7 4 5 1 6 1 4 2 4 4 7 1 3 1 5 3 6	2 1 1 0 1

Замечание

В первых двух примерах запросы спрашивают количество достижимых из каждой башни башен.

В первом примере с 1-й башни можно прыгнуть на башни 1 и 5. Любая другая башня имеет меньшую высоту, чем башня 1, поэтому туда нельзя прыгнуть (в качестве k можно выбрать 1). Множество достижимых из 1-й башни также состоит из башен 1 и 5. Со второй башни можно прыгнуть на башни 1, 2, и 5, они же являются множеством достижимых. С третьей башни можно прыгнуть на башни 2, 3, 5. Однако, башня 1 также является достижимой, поскольку можно сделать два прыжка: $3 \to 2 \to 1$. Таким образом, получается 4 достижимые башни. С 4-й башни можно прыгнуть на башни 4 и 5, они же являются единственными достижимыми. Из 5-й башни достижима только она сама.

Во втором примере из 1-й и из 2-й башни достижимы башни $1, 2, 3, 4, 5$. Из 3-й башни достижимы башни $3, 4, 5$. Из 4-й и 5-й башни достижимы башни $4, 5$. Из 6-й башни достижимы башни $4, 5, 6$. Из 7-й башни достижимы башни $4, 5, 6, 7$.

Рассмотрим третий пример:

- В первом запросе множество индексов башен k на отрезке $[l, r]$, достижимых из u и из v — $\{3, 6\}$.
- Во втором запросе множество индексов башен k на отрезке $[l, r]$, достижимых из u и из v — $\{6\}$.
- В третьем запросе множество индексов башен k на отрезке $[l, r]$, достижимых из u и из v — $\{3\}$.

- В четвёртом запросе множество индексов башен k на отрезке $[l, r]$, достижимых из u и из v пусто.
- В пятом запросе множество индексов башен k на отрезке $[l, r]$, достижимых из u и из v — $\{6\}$.

Система оценки

Тесты к этой задаче состоят из десяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Обозначим за C максимальную высоту башни.

Группа	Баллы	Ограничения				Необходимые группы
		n	C	q	Дополнительно	
0	0				Тесты из условия	
1	9	$n \leqslant 100$	$C \leqslant 10^9$	$q \leqslant 1000$		0
2	6	$n \leqslant 2000$	$C \leqslant 10^9$	$q \leqslant 500\,000$	Все a_i различны	
3	7	$n \leqslant 200\,000$	$C \leqslant 2$	$q \leqslant 100\,000$	$u_i = v_i$, $l_i = 1$, $r_i = n$	
4	11	$n \leqslant 200\,000$	$C \leqslant 3$	$q \leqslant 100\,000$	$u_i = v_i$, $l_i = 1$, $r_i = n$	3
5	15	$n \leqslant 10\,000$	$C \leqslant 10^9$	$q \leqslant 10\,000$	$u_i = v_i$, $l_i = 1$, $r_i = n$	
6	18	$n \leqslant 500\,000$	$C \leqslant 10^9$	$q \leqslant 500\,000$	$u_i = v_i$, $l_i = 1$, $r_i = n$	3–5
7	12	$n \leqslant 500\,000$	$C \leqslant 10^9$	$q \leqslant 500\,000$	$u_i = v_i$	3–6
8	5	$n \leqslant 500\,000$	$C \leqslant 10^9$	$q \leqslant 500\,000$	$r_i \leqslant \min(u_i, v_i)$ или $l_i \geqslant \max(u_i, v_i)$	3–6
9	11	$n \leqslant 500\,000$	$C \leqslant 10^9$	$q \leqslant 500\,000$	$l_i = 1$, $r_i = n$	3–6
10	6	$n \leqslant 500\,000$	$C \leqslant 10^9$	$q \leqslant 500\,000$	**Offline-проверка**	0–9