

1B. Количество максимумов на отрезке

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления значения максимального из нескольких подряд идущих элементов массива, а также количества элементов, равных максимальному на данном отрезке.

Входные данные

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 1 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число K ($1 \leq K \leq 30\,000$) — количество запросов на вычисление максимума.

В следующих K строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

Выходные данные

Для каждого запроса выведите в отдельной строке через пробел значение максимального элемента на указанном отрезке массива и количество максимальных элементов на этом отрезке.

Пример

входные данные	Скопировать
5 2 2 2 1 5 2 2 3 2 5	
выходные данные	Скопировать
2 2 5 1	

1С. Нолики

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дедус любит давать своим ученикам сложные задачки. На этот раз он придумал такую задачу:

Рейтинг всех его учеников записан в массив A . Запросы Дедуса таковы:

1. Изменить рейтинг i -го ученика на число x
2. Найти максимальную последовательность подряд идущих ноликов в массиве A на отрезке $[l, r]$.

Помогите бедным фиксикам избежать зверского наказания за нерешение задачи на этот раз.

Входные данные

В первой строке входного файла записано число N ($1 \leq N \leq 500\,000$) – количество учеников. Во второй строке записано N чисел – их рейтинги, числа по модулю не превосходящие 1000 (по количеству задач, которые ученик решил или не решил за время обучения). В третьей строке записано число M ($1 \leq M \leq 50\,000$) – количество запросов. Каждая из следующих M строк содержит описания запросов:

«UPDATE i x » – обновить i -ый элемент массива значением x ($1 \leq i \leq N$, $|x| \leq 1000$)

«QUERY l r » – найти длину максимальной последовательности из нулей на отрезке с l по r . ($1 \leq l \leq r \leq N$)

Выходные данные

В выходной файл выведите ответы на запросы «QUERY» в том же порядке, что и во входном файле

Пример

входные данные	Скопировать
5 328 0 0 0 0 5 QUERY 1 3 UPDATE 2 832 QUERY 3 3 QUERY 2 3 UPDATE 2 0	
выходные данные	Скопировать
2 1 1	

1D. Катый ноль

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс k -го слева нуля на данном отрезке в массиве.

Входные данные

В первой строке вводится одно натуральное число N ($1 \leq N \leq 200\,000$) — количество чисел в массиве. Во второй строке вводятся N чисел от 0 до $100\,000$ — элементы массива. В третьей строке вводится одно натуральное число M ($1 \leq M \leq 200\,000$) — количество запросов. Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (z — вычислить индекс k -го нуля, u — обновить значение элемента). Следом за z вводится три числа — левый и правый концы отрезка и число k ($1 \leq k \leq N$). Следом за u вводятся два числа — номер элемента и его новое значение.

Выходные данные

Для каждого запроса z выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите -1 для данного запроса.

Пример

входные данные	Скопировать
<pre>5 0 0 3 0 2 3 u 1 5 u 1 0 s 1 5 3</pre>	
выходные данные	Скопировать
<pre>4</pre>	

1E. Ближайшее большее число справа

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Дан массив a из n чисел. Нужно обрабатывать запросы:

- $\text{set}(i, x)$ – присвоить новое значение элементу массива $a[i] = x$;
- $\text{get}(i, x)$ – найти $\min k: k \geq i$ и $a_k \geq x$.

Входные данные

Первая строка входных данных содержит два числа: длину массива n и количество запросов m ($1 \leq n, m \leq 200\,000$).

Во второй строке записаны n целых чисел – элементы массива a ($0 \leq a_i \leq 200\,000$).

Следующие m строк содержат запросы, каждый запрос содержит три числа t, i, x . Первое число t равно 0 или 1 – тип запроса. $t = 0$ означает запрос типа set , $t = 1$ соответствует запросу типа get , $1 \leq i \leq n$, $0 \leq x \leq 200\,000$. Элементы массива нумеруются с единицы.

Выходные данные

На каждой запрос типа get на отдельной строке выведите соответствующее значение k . Если такого k не существует, выведите -1 .

Пример

входные данные	Скопировать
4 5 1 2 3 4 1 1 1 1 1 3 1 1 5 0 2 3 1 1 3	
выходные данные	Скопировать
1 3 -1 2	

1Н. Сережа и скобочки

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

У Сережи есть строка s длины n , состоящая из символов «(» и «)».

Сереже нужно ответить на m запросов, каждый из которых характеризуется двумя целыми числами l_i, r_i . Ответом на i -ый запрос является длина наибольшей правильной скобочной подпоследовательности последовательности $s_{l_i}, s_{l_i+1}, \dots, s_{r_i}$. Помогите Сереже ответить на все запросы.

Входные данные

Первая строка содержит последовательность символов без пробелов s_1, s_2, \dots, s_n ($1 \leq n \leq 10^6$). Каждый символ это либо «(», либо «)». Вторая строка содержит целое число m ($1 \leq m \leq 10^5$) количество запросов. Каждая из следующих m строк содержит пару целых чисел. В i -ой строке записаны числа l_i, r_i , ($1 \leq l_i \leq r_i \leq n$) — описание i -го запроса.

Выходные данные

Выведите ответ на каждый запрос в отдельной строке. Ответы выводите в порядке следования запросов во входных данных.

Пример

входные данные	Скопировать
<pre>()()()()() 7 1 1 2 3 1 2 1 12 8 12 5 11 2 10</pre>	
выходные данные	Скопировать
<pre>0 0 2 10 4 6 6</pre>	

Примечание

Подпоследовательностью длины $|x|$ строки $s = s_1 s_2 \dots s_{|s|}$ (где $|s|$ — длина строки s) называется строка $x = s_{k_1} s_{k_2} \dots s_{k_{|x|}}$ ($1 \leq k_1 < k_2 < \dots < k_{|x|} \leq |s|$).

Правильной скобочной последовательностью называется скобочная последовательность, которую можно преобразовать в корректное арифметическое выражение путем вставок между ее символами символов «1» и «+». Например, скобочные последовательности «()()», «(())» — правильные (полученные выражения: «(1)+(1)», «((1+1)+1)»), а «)(» и «(» — нет.

Для третьего запроса искомая последовательность будет «()».

Для четвертого запроса искомая последовательность будет «()()()()».

2A. Максимум на подотрезках с добавлением на отрезке

ограничение по времени на тест: 0.5 секунд

ограничение по памяти на тест: 256 мегабайт

Реализуйте эффективную структуру данных для хранения массива и выполнения следующих операций: увеличение всех элементов данного интервала на одно и то же число; поиск максимума на интервале.

Входные данные

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 — элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30000$) — количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (m — найти максимум, a — увеличить все элементы на отрезке).

Следом за m вводятся два числа — левая и правая граница отрезка.

Следом за a вводятся три числа — левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($0 \leq add \leq 100000$).

Выходные данные

Выведите в одну строку через пробел ответы на каждый запрос m .

Пример

входные данные

Скопировать

```
5
2 4 3 1 5
5
m 1 3
a 2 4 100
m 1 3
a 5 5 10
m 1 5
```

выходные данные

Скопировать

```
4 104 104
```


2F. Окона

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

Входные данные

В первой строке входного файла записано число окон n ($1 \leq n \leq 50\,000$).

Следующие n строк содержат координаты окон $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$, где $(x_{i,1}, y_{i,1})$ — координаты левого верхнего угла i -го окна, а $(x_{i,2}, y_{i,2})$ — правого нижнего (на экране компьютера y растет сверху вниз, а x — слева направо). Все координаты — целые числа, по модулю не превосходящие 10^6 .

Выходные данные

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенных пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

Примеры

входные данные	Скопировать
2 0 0 3 3 1 1 4 4	
выходные данные	Скопировать
2 1 3	
входные данные	Скопировать
1 0 0 1 1	
выходные данные	Скопировать
1 0 1	
входные данные	Скопировать
4 0 0 1 1 0 1 1 2 1 0 2 1 1 1 2 2	
выходные данные	Скопировать
4 1 1	

3A. Разреженные таблицы

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Входные данные

В первой строке входного файла даны три натуральных числа n , m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^7$) и a_1 ($0 \leq a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leq u_1, v_1 \leq n$) — первый запрос.

Элементы a_2, a_3, \dots, a_n задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

$$\begin{aligned} u_{i+1} &= ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1, \\ v_{i+1} &= ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1, \end{aligned}$$

где ans_i — ответ на запрос номер i .

Обратите внимание, что u_i может быть больше, чем v_i .

Выходные данные

В выходной файл выведите u_m , v_m и ans_m (последний запрос и ответ на него).

Пример

входные данные	Скопировать
10 8 12345 3 9	
выходные данные	Скопировать
5 3 1565158	

3B. Звёзды

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вася любит наблюдать за звёздами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером $n \times n \times n$. Этот куб поделен на маленькие кубики размером $1 \times 1 \times 1$. Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звёзд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

Входные данные

Первая строка входного файла содержит натуральное число $1 \leq n \leq 128$. Координаты кубиков — целые числа от 0 до $n - 1$. Далее следуют записи о происходивших событиях по одной в строке. В начале строки записано число m . Если m равно:

1. Тогда за ним следуют 4 числа — x, y, z ($0 \leq x, y, z < N$) и k ($-20000 \leq k \leq 20000$) — координаты кубика и величина, на которую в нем изменилось количество видимых звёзд;
2. Тогда за ним следуют 6 чисел — $x_1, y_1, z_1, x_2, y_2, z_2$ ($0 \leq x_1 \leq x_2 < N, 0 \leq y_1 \leq y_2 < N, 0 \leq z_1 \leq z_2 < N$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$;
3. Это означает, что Васе надоело наблюдать за звёздами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней.

Количество записей во входном файле не больше 100 002.

Выходные данные

Для каждого Петиного вопроса выведите искомое количество звёзд.

Пример

входные данные	Скопировать
2 2 1 1 1 1 1 1 1 0 0 0 1 1 0 1 0 3 2 0 0 0 0 0 0 2 0 0 0 0 1 0 1 0 1 0 -2 2 0 0 0 1 1 1 3	
выходные данные	Скопировать
0 1 4 2	

4A. Set

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Реализуйте множество с использованием хеш-таблицы.

В данной задаче запрещено пользоваться стандартной библиотекой языка (классами `std::set`, `std::unordered_set` и подобными в C++ и их аналогами в других языках программирования).

Входные данные

Входные данные содержат описание операций, которые необходимо выполнить. Каждая строка содержит одну из следующих операций:

- `insert x` — добавить элемент x в множество. Если элемент уже есть в множестве, не нужно ничего делать.
- `delete x` — удалить элемент x из множества. Если элемента нет в множестве, не нужно ничего делать.
- `exists x` — если элемент x есть в множестве, необходимо вывести «true». В противном случае необходимо вывести «false».

Все числа, используемые в запросах целые и не превосходят 10^9 по модулю.

Выходные данные

Для каждой операции `exists` выведите результат ее работы.

Пример

входные данные	Скопировать
<pre>insert 2 insert 5 insert 3 exists 2 exists 4 insert 2 delete 2 exists 2</pre>	
выходные данные	Скопировать
<pre>true false false</pre>	

4B. LinkedHashMap

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Реализуйте LinkedHashMap с использованием хеш-таблицы.

В данной задаче запрещено пользоваться стандартной библиотекой языка (классами `std::map`, `std::unordered_map` и подобными в C++ и их аналогами в других языках программирования).

Входные данные

Входные данные содержат описание операций, которые необходимо выполнить. Каждая строка содержит одну из следующих операций:

- `put x y` — добавить в соответствие ключу x значение y . Если такой ключ уже есть, значение нужно изменить на y .
- `delete x` — удалить ключ x . Если элемента нет, не нужно ничего делать.
- `get x` — если ключ x есть, необходимо вывести соответствующее ему значение. В противном случае необходимо вывести «none».
- `prev x` — вывести значение, соответствующее ключу, который был вставлен позже всех, но до ключа x . Если такого ключа нет, или ключа x нет, необходимо вывести «none».
- `next x` — вывести значение, соответствующее ключу, который был вставлен раньше всех, но после ключа x . Если такого ключа нет, или ключа x нет, необходимо вывести «none».

Все ключи и значения — строки из латинских букв длиной не более 20 символов.

Выходные данные

Для каждой операции `get`, `prev` и `next` выведите результат ее работы.

Пример

входные данные

[Скопировать](#)

```
put zero a
put one b
put two c
put three d
put four e
get two
prev two
next two
delete one
delete three
get two
prev two
next two
next four
```

выходные данные

[Скопировать](#)

```
c
b
d
c
a
e
none
```

4D. Хеш-код

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Согласно документации стандартной библиотеки Java, хеш-код для строки вычисляется как:

$$s[0] \cdot 31^{n-1} + s[1] \cdot 31^{n-2} + \dots + s[n-1]$$

Где $s[i]$ — это i -й символ строки, n длина строки. Для вычисления используются целые 32-битные числа в форме дополнения до двух.

Вы собираетесь взломать сервера одной известной компании. Чтобы вы смогли выполнить атаку, вам нужны k различных строк, которые имеют одинаковые хеш-коды. К сожалению, сервера этой компании не принимают строки запроса, содержащие буквы отличные от английских букв нижнего и верхнего регистров.

Напишите программу, которая генерирует такие строки.

Входные данные

Первая строка содержит целое число k — количество необходимых строк запроса для генерации ($2 \leq k \leq 1000$).

Выходные данные

Необходимо вывести k различных непустых строк, каждая из которых имеет длину не более 1000 символов. Все строки должны состоять только из английских букв верхнего или нижнего регистров и иметь равный хеш-код.

Пример

входные данные	Скопировать
4	
выходные данные	Скопировать
edHs mENAGeS fEHs edIT	

5A. И снова сумма...

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Входные данные

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «+ i », либо «? l r ». Операция «? l r » задает запрос $sum(l, r)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $add(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Выходные данные

Для каждого запроса выведите одно число — ответ на запрос.

Пример

входные данные	Скопировать
<pre>6 + 1 + 3 + 3 ? 2 4 + 1 ? 2 4</pre>	
выходные данные	Скопировать
<pre>3 7</pre>	

5B. Вперёд!

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

{256 мегабайт}

Капрал Дукар любит раздавать приказы своей роте. Самый любимый его приказ — «Вперёд!». Капрал строит солдат в ряд и отдаёт некоторое количество приказов, каждый из которых звучит так: «Рядовые с l_i по r_i — вперёд!»

Перед тем, как Дукар отдал первый приказ, солдаты были пронумерованы от 1 до n слева направо. Услышав приказ «Рядовые с l_i по r_i — вперёд!», солдаты, стоящие на местах с l_i по r_i включительно, продвигаются в начало ряда в том же порядке, в котором были.

Например, если в какой-то момент солдаты стоят в порядке 2, 3, 6, 1, 5, 4, то после приказа «Рядовые с 2 по 4 — вперёд!», порядок будет таким: 3, 6, 1, 2, 5, 4. А если потом Капрал вышлет вперёд солдат с 3 по 4, то порядок будет уже таким: 1, 2, 3, 6, 5, 4.

Вам дана последовательность приказов Капрала. Найдите порядок, в котором будут стоять солдаты после исполнения всех приказов.

Входные данные

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — число солдат и число приказов. Следующие m строк содержат приказы в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Выходные данные

Выведите в выходной файл n целых чисел — порядок, в котором будут стоять солдаты после исполнения всех приказов.

Пример

входные данные	Скопировать
6 3 2 4 3 5 2 2	
выходные данные	Скопировать
1 4 5 2 3 6	

6A. LCA Problem

ограничение по времени на тест: 5 секунд

ограничение по памяти на тест: 256 мегабайт

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 10^5$) вершин, пронумерованных от 0 до $n - 1$. Требуется ответить на m ($1 \leq m \leq 10^6$) запросов о наименьшем общем предке для пары вершин.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y, z . Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид (a_1, a_2) . Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $(a_{2i-1} + v) \bmod n, a_{2i})$.

Входные данные

Первая строка содержит два числа: n и m . Корень дерева имеет номер 0.

Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру родителя вершины i .

Третья строка содержит два целых числа в диапазоне от 0 до $n - 1$: a_1 и a_2 .

Четвертая строка содержит три целых числа: x, y, z , эти числа неотрицательны и не превосходят 10^9 .

Выходные данные

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

Примеры

входные данные	Скопировать
<pre>3 2 0 1 2 1 1 1 0</pre>	
выходные данные	Скопировать
<pre>2</pre>	

входные данные	Скопировать
<pre>1 2 0 0 1 1 1</pre>	
выходные данные	Скопировать
<pre>0</pre>	

6C. Самое дешевое ребро

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на M запросов вида "найти у двух вершин минимум среди стоимостей ребер пути между ними".

Входные данные

В первой строке файла записано одно числ — n (количество вершин).

В следующих $n - 1$ строках записаны два числа — x и y . Число x на строке i означает, что x — предок вершины i , y означает стоимость ребра.

$x < i$, $|y| \leq 10^6$.

Далее m запросов вида (x, y) — найти минимум на пути из x в y ($x \neq y$).

Ограничения: $2 \leq n \leq 5 \cdot 10^4$, $0 \leq m \leq 5 \cdot 10^4$.

Выходные данные

Выведите m ответов на запросы.

Пример

входные данные	Скопировать
5 1 2 1 3 2 5 3 2 2 2 3 4 5	
выходные данные	Скопировать
2 2	

7B. Прибавление на пути

ограничение по времени на тест: 4 секунды
ограничение по памяти на тест: 256 мегабайт

Задано дерево. В каждой вершине есть значение, изначально все значения равны нулю. Требуется обработать запрос прибавления на пути и запрос значения в вершине.

Входные данные

В первой строке задано целое число n — число вершин в дереве ($1 \leq n \leq 3 \cdot 10^5$).

В следующих $n - 1$ строках заданы ребра дерева: по два целых числа v и u в строке — номера вершин, соединенных ребром ($1 \leq v, u \leq n$).

В следующей строке задано целое число m — число запросов ($1 \leq m \leq 5 \cdot 10^5$).

Следующие m строк содержат запросы в одном из двух форматов:

- $+ \ v \ u \ d$ — прибавить число d во все значения в вершинах на пути от v до u ($1 \leq v, u \leq n$; $1 \leq d \leq 10^9$);
- $? \ v$ — вывести значение в вершине v ($1 \leq v \leq n$).

Выходные данные

Выведите ответы на все запросы.

Пример

входные данные	Скопировать
5 1 2 1 3 3 4 3 5 5 + 2 5 1 ? 3 + 1 1 2 ? 1 ? 3	
выходные данные	Скопировать
1 3 1	

1A. RMQ

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Реализуйте структуру данных, которая на данном массиве из N целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

Входные данные

В первой строке вводится натуральное число N ($1 \leq N \leq 10^5$) – количество элементов в массиве. В следующей строке содержатся N целых чисел, не превосходящих по модулю 10^9 – элементы массива. Гарантируется, что в массиве нет одинаковых элементов. Далее идет число K ($0 \leq K \leq 10^5$) – количество запросов к структуре данных. Каждая из следующих K строк содержит два целых числа l и r ($1 \leq l \leq r \leq N$) – левую и правую границы отрезка в массиве для данного запроса.

Выходные данные

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от l до r и индекс одного из элементов массива, принадлежащий отрезку от l до r , на котором достигается этот максимум.

Пример

входные данные	Скопировать
5 7 3 1 6 4 3 1 5 2 4 3 3	
выходные данные	Скопировать
7 1 6 4 1 3	