

Задача А. Отрезок с максимальной суммой

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

Дан массив целых чисел. Найти отрезок этого массива с максимальной суммой.

Формат входных данных

В первой строке дано натуральное число n ($1 \leq n \leq 100\,000$) — размер массива. Во второй строке через пробел перечислены элементы массива. Числа по модулю не превышают 10 000.

Формат выходных данных

Выведите три числа — индекс начала отрезка, индекс конца и саму максимальную сумму. Отрезок не может быть пустым, но отрезок из одного элемента не считается пустым. Массив индексируется с единицы. Если ответов несколько — выведите любой.

Примеры

<code>max-segment.in</code>	<code>max-segment.out</code>
5 -1 2 3 -2 5	2 5 8

Задача В. Пупа и Лупа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

В бухгалтерии хранится массив размера n , содержащий целочисленные элементы. Лупе поручили ответственное задание. Ему нужно найти минимум на m отрезках фиксированной длины k на этом массиве. Чтобы Пупе не пришлось выполнять за своего друга работу, помогите Лупе справиться с этой задачей.

Формат входных данных

В первой строке вводятся три числа n, m, k ($1 \leq n, m, k, \leq 1e6$), $k \leq n$. Далее вводится массив размера n ($1 \leq a_i \leq 1e9$). После этого в m строках вводятся пары чисел $1 \leq l \leq r, \leq n$

Формат выходных данных

Выведите m чисел на m строках - ответы на задачу.

Примеры

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

Задача С. Очень скоростной трамвай

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Организаторы чемпионата решили, что добираться до стадиона болельщики будут на трамвае. Для этого существующий трамвайный маршрут, состоящий из n остановок, продлили и дополнили ещё одной остановкой «Стадион».

Однако выяснилось, что поездка на трамвае занимает немало времени. Мэру города S пришла в голову отличная идея: сократить количество трамвайных остановок. Проведя некоторые подсчёты, он решил оставить ровно две остановки: новую — «Стадион» и одну из имеющихся n остановок.

По заданию мэра специалисты по транспорту провели исследование и выяснили, что в настоящее время трамвайным маршрутом пользуется m человек. Для каждого из этих m человек известно, на какой остановке s_j он садится в трамвай и на какой остановке f_j выходит из трамвая.

Когда из имеющихся n остановок останется одна, пассажиру, который пожелает воспользоваться трамваем, придётся пешком дойти от остановки, где он обычно садился в трамвай, до оставшейся остановки, доехать до остановки «Стадион», а затем дойти от остановки «Стадион» до остановки, где он обычно выходил.

Конечно, пассажир может переквалифицироваться в пешехода и просто дойти пешком от остановки, где он обычно садился, до остановки, где он обычно выходил, т.е. проделать путь длиной в $f_j - s_j$ остановок.

Мэр полагает, что если путь длиной в $f_j - s_j$ остановок окажется меньше, нежели пешеходная часть пути с использованием трамвая, пассажир совершенно точно пойдёт пешком.

Мэр ещё не решил, какую из имеющихся остановок он хочет оставить. У него есть список вариантов, и для каждого из них он хотел бы знать, какое количество пассажиров воспользуется трамваем. Ваша задача — определить это количество для каждого номера остановки из списка мэра.

Формат входных данных

В первой строке содержатся целые числа n, m, q ($2 \leq n \leq 3 \cdot 10^5$, $1 \leq m \leq 3 \cdot 10^5$, $1 \leq q \leq 3 \cdot 10^5$) — количество остановок в маршруте, количество пассажиров, пользующихся маршрутом и количество запросов.

Во второй строке содержится m целых чисел s_1, s_2, \dots, s_m ($1 \leq s_j \leq n - 1$, $j = 1, 2, \dots, m$), где s_j — номер остановки, на которой пассажир $\#j$ обычно садится в трамвай.

В третьей строке содержится m целых чисел f_1, f_2, \dots, f_m ($2 \leq f_j \leq n$, $j = 1, 2, \dots, m$), где f_j — номер остановки, на которой пассажир $\#j$ обычно выходит из трамвая. Гарантируется, что $s_j < f_j$ для всех j .

В четвёртой строке содержится q целых чисел r_1, r_2, \dots, r_q ($1 \leq r_k \leq n$, $k = 1, 2, \dots, q$), где r_k — номер оставшейся остановки, для которой нужно вычислить количество пассажиров, воспользовавшихся трамваем.

Формат выходных данных

Выведите q чисел p_1, p_2, \dots, p_k .

Число p_k ($1 \leq k \leq q$) — это количество пассажиров, которые воспользуются трамваем, если единственной оставшейся из имеющихся остановок будет остановка $\#r_k$.

При выводе разделяйте числа пробелами или переводами строк.

Примеры

стандартный ввод	стандартный вывод
5 14 4	6 5 3 6
2 1 3 3 4 2 1 2 2 1 3 3 4 2	
4 3 4 5 5 3 3 5 4 3 5 4 5 4	
2 4 3 2	

Задача D. Макс и префиксные матрицы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Недавно Макс начал использовать на занятиях с учениками новый сборник учебных задач по программированию. Чтобы иметь представление о том, насколько успешно ученики справляются с задачами, Макс решил составить специальную матрицу S .

У Макса N учеников, каждый из которых имеет идентификатор от 0 до $(N - 1)$. Сборник содержит M задач, каждая из которых имеет идентификатор от 0 до $(M - 1)$. Если ученик с идентификатором i решил задачу с идентификатором j , то на пересечении i -й строки и j -го столбца матрицы S находится единица ($S[i][j] = 1$), а иначе — ноль ($S[i][j] = 0$).

Кроме того, Макс составил префиксную матрицу P . Элемент $P[i][j]$ равен сумме всех элементов $S[i_1][j_1]$, для которых справедливы неравенства $1 \leq i_1 \leq i$ и $1 \leq j_1 \leq j$.

Сегодня Макс рассказывал Владимиру об успехах своих учеников. Владимир поинтересовался, сколько решений отправили ученики, идентификаторы которых принадлежат диапазону $[Y_1; Y_2]$, по задачам, идентификаторы которых принадлежат диапазону $[X_1; X_2]$. Макс собрался было показать Владимиру матрицу S , но обнаружил, что она бесследно исчезла!

Помогите Максиму восстановить ответы на вопросы Владимира, используя только матрицу P .

Формат входных данных

Первая строка содержит целые числа N и M ($1 \leq N, M \leq 100$) — количество учеников и задач соответственно.

Следующие N строк описывают матрицу P . Каждая из них содержит M целых чисел P_{ij} ($0 \leq P_{ij} \leq 10^4$) — элементы матрицы.

Следующая строка содержит целое число Q ($1 \leq Q \leq 10^6$) — количество запросов Владимира.

Следующие Q строк описывают запросы. Каждая из них содержит целые числа $Y_{1k}, Y_{2k}, X_{1k}, X_{2k}$ ($0 \leq Y_{1k} \leq Y_{2k} \leq N - 1, 0 \leq X_{1k} \leq X_{2k} \leq M - 1$) — соответственно границы диапазона учеников и границы диапазона задач.

Формат выходных данных

Для каждого запроса выведите на отдельной строке одно целое число — количество решений, которое отправили ученики с идентификаторами в диапазоне $[Y_{1k}; Y_{2k}]$ по задачам с идентификаторами в диапазоне $[X_{1k}; X_{2k}]$.

Примеры

стандартный ввод	стандартный вывод
1 5 1 2 2 3 3 3 0 0 0 0 0 0 2 2 0 0 3 4	1 0 1
3 4 0 1 1 2 1 3 4 5 2 4 6 8 5 0 2 0 3 0 0 0 3 1 1 0 3 2 2 0 3 1 2 1 2	8 2 3 3 3

Задача Е. Разрезанные таблицы

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

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

Формат входных данных

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:

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

где ans_i — ответ на запрос номер i .

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

Формат выходных данных

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

Примеры

<code>sparse.in</code>	<code>sparse.out</code>
10 8 12345 3 9	5 3 1565158

Замечание

Пояснение к тесту из примера: запросы и результаты.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
12345	305498	7048017	11694653	1565158	2591019	9471233	570265	13137658	1325095

#	u	v	ans
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

Задача F. Дерево отрезков с операцией НОД

Имя входного файла: `segment-gcd.in`
Имя выходного файла: `segment-gcd.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления НОД нескольких подряд идущих элементов массива.

Формат входных данных

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 НОД.

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

Формат выходных данных

Для каждого запроса выведите НОД всех чисел соответствующего участка массива. Числа выводите в одну строку через пробел.

Примеры

<code>segment-gcd.in</code>	<code>segment-gcd.out</code>
5	2
2 2 2 1 5	1
2	
2 3	
2 5	

Замечание

TL для Python 3 секунды

Задача G. Быстрое прибавление

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Есть массив целых чисел длины $n = 2^{24}$, изначально заполненных нулями. Вам нужно сперва обработать m случайных запросов вида “прибавление на отрезке”. Затем обработать q случайных запросов вида “сумма на отрезке”.

Формат входных данных

На первой строке числа m, q ($1 \leq m, q \leq 2^{24}$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
0. unsigned int a, b; // даны во входных данных
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до 2^24-1.
5. }
```

Каждый запрос первого вида генерируется следующим образом:

```
1. add = nextRand(); // число, которое нужно прибавить
2. l = nextRand();
3. r = nextRand();
4. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Каждый запрос второго вида генерируется следующим образом:

```
1. l = nextRand();
2. r = nextRand();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Сперва генерируются запросы первого вида, затем второго.

Формат выходных данных

Выведите сумму ответов на все запросы второго типа по модулю 2^{32} .

Примеры

стандартный ввод	стандартный вывод
5 5 13 239	811747796

Замечание

Последовательность запросов в тесте из примера:

```
[13..170] += 0
[28886..375523] += 2221
[2940943..13131777] += 4881801
[2025901..10480279] += 4677840
[4943766..6833065] += 9559505
get sum [13412991..13937319]
get sum [1871500..6596736]
get sum [7552290..14293694]
get sum [1268651..16492476]
get sum [2210673..13075602]
```

Задача Н. Поиск максимума

Имя входного файла: `index-max.in`
Имя выходного файла: `index-max.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления номера максимального из нескольких подряд идущих элементов массива.

Формат входных данных

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).

Формат выходных данных

Для каждого запроса выведите индекс максимального элемента на указанном отрезке массива. Если максимальных элементов несколько, выведите любой их них.

Числа выводите в одну строку через пробел.

Примеры

<code>index-max.in</code>	<code>index-max.out</code>
5	3 5
2 2 2 1 5	
2	
2 3	
2 5	

Замечание

TL для Python 2 секунды

Задача I. Сигнальные башни

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Миша и Маша решили почитать учебник истории вместо учебника по информатике. Из него они узнали про византийскую систему сигнальных башен, которая им очень понравилась.

В Византии было m разных чертежей башен, i -я башня имеет высоту i и красоту b_i по мнению византийцев. При этом, чтобы система работала, наблюдатель, сидящий в каждой башне и поджигающий огонь, должен видеть огонь предыдущей башни. Каждый наблюдатель смотрит только налево и имеет угол обзора в 90° , под углом 45° к земле. Иначе говоря, более левый наблюдатель видит более правого, если абсолютная разница между высотами их башен не превосходит абсолютного расстояния между основаниями их башен.

Маша очень любит красивые конструкции и ей стало интересно, какова может быть максимальная суммарная красота сигнальных башен, если строить их можно только в одну линию в целочисленных координатах от 1 до n . При этом в нулевой координате уже построена старая некрасивая башня (её красота равна нулю) высоты x .

Помогите Мише впечатлить Машу и решить эту непростую задачу.

Формат входных данных

Первая строка входных данных содержит три натуральных числа n , m и x ($1 \leq n \leq 100, 1 \leq m \leq 1000, 1 \leq x \leq m$). Вторая строка содержит m натуральных чисел b_i ($1 \leq b_i \leq 10^9$).

Формат выходных данных

Программа должна вывести одно число — максимально возможную красоту конструкции.

Примеры

стандартный ввод	стандартный вывод
5 8 3 4 6 2 1 3 5 7 8	30
5 8 3 6 4 3 2 7 8 5 1	33

Задача J. Интеллектуальный отпуск

Имя входного файла:	e.in
Имя выходного файла:	e.out
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Туристическая отрасль в этом сезоне столкнулась с серьёзными сложностями. Добросовестные туроператоры ищут новые рекламные ходы для продажи своих туров. Как известно, наиболее благоприятная для отдыха погода меняется плавно, причём не только от одного дня к другому, но и в течение суток.

Для большинства туристических направлений есть многолетние посекундные результаты измерений различных климатических параметров, например, температуры или влажности. У каждого человека своё понимание того, насколько различными могут быть подобные значения во время отпуска, но всех интересуют непрерывные туры как можно большей продолжительности.

Пусть мы зафиксировали туристическое направление и некоторый климатический параметр. Будем называть изменчивостью тура разницу между максимальным и минимальным значением выбранного параметра за всё время поездки. Для каждого туриста известно максимальное приемлемое значение изменчивости k_i .

Даны результаты измерений некоторого климатического параметра на одном из курортов и значения k_i для нескольких туристов. Требуется для каждого из них определить максимальный диапазон, подходящий для отпуска.

Формат входных данных

В первой строке входного файла находится целое число N ($1 \leq N \leq 600\,000$) — количество сделанных измерений. Во второй строке — N целых чисел, по модулю не превосходящих 10^9 — данные посекундных измерений.

В третьей строке входного файла находится число M ($1 \leq M \leq 100$) — количество туристов, для которых необходимо найти оптимальный диапазон. В четвёртой строке — M целых чисел k_1, k_2, \dots, k_M ($0 \leq k_i \leq 10^9$) — максимальная возможная разница между выбранным климатическим параметром в непрерывном диапазоне дней для каждого из туристов.

Формат выходных данных

В выходной файл для каждого из M запросов в отдельной строке выведите два числа: номер первого измерения диапазона и номер последнего измерения, входящего в диапазон. Нумерация измерений ведётся с единицы. Если для некоторого туриста существует несколько подходящих диапазонов максимальной длины, выведите границы любого из них.

Примеры

e.in	e.out
7	3 5
10 1 10 12 11 1 11	4 5
2	
2 1	
9	3 4
1 5 2 3 6 4 7 8 9	1 9
6	7 9
1 10 2 4 5 0	2 6
	1 6
	1 1

Задача К. Друзья и последовательности

Имя входного файла: friends.in
Имя выходного файла: friends.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Майк и !Майк соперничают еще со школьных лет, они противоположны во всем что делают, кроме программирования. Сегодня у них возникла проблема, которую сами друзья сами решить не могут, но вместе с вами — кто знает?

Каждый из них знает две последовательности n чисел a и b . По запросу в виде пары целых чисел (l, r) Майк может сразу сообщить значение $\max_{i=l}^r a_i$, а !Майк — значение $\min_{i=l}^r b_i$.

Предположим, что робот задает им каждый из возможных различных запросов в виде пары целых чисел (l, r) ($1 \leq l \leq r \leq n$) (то есть он сделает ровно $n(n+1)/2$ запросов) и считает, сколько раз их ответы на один и тот же запрос совпадают, то есть для скольких пар выполняется $\max_{i=l}^r a_i = \min_{i=l}^r b_i$.

Сколько случаев совпадения посчитает робот?

Формат входных данных

В первой строке содержится единственное целое число n ($1 \leq n \leq 200\,000$).

Во второй строке содержатся n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — элементы последовательности a .

В третьей строке содержатся n целых чисел b_1, b_2, \dots, b_n ($-10^9 \leq b_i \leq 10^9$) — элементы последовательности b .

Формат выходных данных

Выведите одно целое число — количество совпадений ответов, которые посчитает робот, то есть для скольких пар выполняется $\max_{i=l}^r a_i = \min_{i=l}^r b_i$.

Примеры

friends.in	friends.out
6 1 2 3 2 1 4 6 7 1 2 3 2	2
3 3 3 3 1 1 1	0

Задача L. Красивые боевые заклинания

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

Артём изучает заклинания в Летней Колдовской Школе. Боевым заклинанием является любая строка, каждый символ которой, встречается в ней одинаковое количество раз.

На доске появилась длинная строка s из строчных латинских букв. Артём хочет узнать, сколько подстрок s , таких что множества различных символов в s и в подстроке совпадают, являются боевыми заклинаниями.

Формат входных данных

Единственная строка входных данных содержит строку s длины n . ($1 \leq n \leq 2 \cdot 10^5$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
abccabab	5