

Задача А. Где же мое призвание

Имя входного файла: `mission.in`
Имя выходного файла: `mission.out`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайта

n сурикатов стоят строем по возрастанию роста. Нам известен рост каждого из них. Некоторые из них зарываются в землю, некоторые забираются на холмики, ростом же считается расстояние от земли до затылка суриката, так что у некоторых рост может быть отрицательным. Так они стоят k дней. Каждый день к ним приходит один сурок. Рост сурка измеряется и сурок интересуется, на какое место в ряд его бы поставили (при этом его туда, разумеется, не ставят, он же сурок, в конце концов, а интересуется он лишь из праздного любопытства). Если его рост совпадает с ростом одного или нескольких сурикатов, то сурка интересует наименьшая возможная позиция. Помогите сурикатам отвечать на глупые вопросы сурков. Кстати, сурикаты нумеруют себя с нуля.

Формат входных данных

В первой строчке входного файла записаны два числа: размер n и k ($1 \leq n, k \leq 100001$). Во второй строке записано n целых чисел – рост каждого суриката в порядке строя. В третьей строке записаны k целых чисел – рост каждого пришедшего сурка. Все заданные числа по модулю не превосходят 10^9 .

Формат выходных данных

Для каждого из k сурков выведите соответствующие позиции в строю (каждую в отдельной строке).

Примеры

<code>mission.in</code>	<code>mission.out</code>
5 5	1
1 3 5 7 9	2
2 4 8 1 6	4
	0
	3

Задача В. Слишком умные школьники и сортировка

Имя входного файла: `goodsolvers.in`
Имя выходного файла: `goodsolvers.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 4 мегабайта

В одной смене ЛКШ все без исключения школьники (в количестве m человек) очень хорошо решали задачи. А именно: каждый не решил не больше 1000 задач, а всего задач было ровно n ($1000 \leq n \leq 10^{18}$).

В те времена в ЛКШ не было сводной таблицы, в которой школьники упорядочивались по количеству решённых задач. Недавно вашим преподавателям стало интересно: а сколь-

ко задач было решено у 1-го, 2-го, ..., m -го места. В образовательных целях удовлетворить любопытство преподавателей поручено именно Вам.

Формат входных данных

В первой строке записаны целые положительные числа n и m ($1000 \leq n \leq 10^{18}$, $1 \leq m \leq 2 \cdot 10^6$). Во второй строке записаны числа a_1, \dots, a_m - количество задач, решённых школьниками ($n - 1000 \leq a_i \leq n$).

Формат выходных данных

Выведите в единственной строке m чисел - количество задач, решённых школьниками, занявшими 1-е, 2-е, ..., m -е место, соответственно.

Примеры

<code>goodsolvers.in</code>	<code>goodsolvers.out</code>
1000 3	3 2 1
2 1 3	

Задача С. Слияние последовательностей

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

Даны две бесконечные неубывающие последовательности A и B . Требуется найти k -ый элемент в неубывающей последовательности C , содержащей все элементы из A и B (включая повторы).

Последовательность A задается с помощью полинома $P(x) = x^3$:

$$a_1 = P(1) \bmod 12345, \quad a_i = a_{i-1} + (P(i) \bmod 12345) \text{ при } i > 1$$

Последовательность B задается с помощью полинома $Q(x) = x^2$:

$$b_1 = Q(1) \bmod 123, \quad b_i = b_{i-1} + (Q(i) \bmod 123) \text{ при } i > 1$$

Формат входных данных

Входной файл содержит натуральное число k ($1 \leq k \leq 10^7$).

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу. Гарантируется, что ответ не превышает $2 \cdot 10^9$.

Примеры

<code>merge.in</code>	<code>merge.out</code>
5	14

Задача D. Быстрая сортировка

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

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью алгоритма быстрой сортировки (qsort).

Формат входных данных

В первой строке входного файла содержится число N ($1 \leq N \leq 100\,000$) — количество элементов в массиве. Во второй строке находятся N целых чисел, по модулю не превосходящих 10^9 .

Формат выходных данных

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Примеры

sort.in	sort.out
10	1 1 2 2 3 3 4 6 7 8
1 8 2 1 4 7 3 2 3 6	

Замечание

В этой задаче обязательно использовать быструю сортировку.

Задача E. Инвентаризация

Имя входного файла: `sort.in`
Имя выходного файла: `sort.out`
Ограничение по времени: 8 секунды
Ограничение по памяти: 64 мегабайта

В связи с модернизацией производства на заводе зубных щеток в Тау Кита было решено переписать список роботов, обслуживающих завод. Каждый робот имеет 2 номера: основной и вспомогательный. Новый список должен удовлетворять следующим правилам:

1. Если один робот в новом списке находится раньше другого, то основной номер первого меньше или равен основному номеру второго.
2. Если основные номера роботов равны, то они расположены в таком же порядке, как и в исходном списке.

Тау Китаю обратились к Вам с просьбой переписать список. Помогите модернизации организаций!

Формат входных данных

В первой строке входного файла содержится число N ($1 \leq N \leq 100\,000$) — количество роботов на заводе. На каждой следующей строке находятся 2 числа — основной и вспо-

могательный номера очередного робота. Оба номера неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите N строчек, i -ая содержит 2 числа — основной и вспомогательный номер i -го робота в новом списке.

Примеры

sort.in	sort.out
10	1 8
1 8	1 11
8 9	2 10
2 10	2 23
1 11	3 11
4 2	3 3
7 2	4 2
3 11	6 7
2 23	7 2
3 3	8 9
6 7	

Задача F. Футбольный турнир

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

В ЛКШ прошел турнир по футболу. Теперь физику Саше необходимо подвести результаты и составить итоговую таблицу. Турнир прошел в один круг, т. е. каждая команда сыграла с каждой ровно один раз. За победу команда получила 3 очка, за ничью 1 очко, а за поражение 0 очков. Кроме того в таблице учитывается разница пропущенных и забитых мячей. Требуется упорядочить команды по невозрастанию набранных очков, а при равенстве очков раньше в таблице должна идти команда, у которой разница забитых и пропущенных мячей больше.

Формат входных данных

В первой строке вводится единственное число N ($1 \leq N \leq 100\,000$) — количество команд, участвовавших в турнире. Во второй строке записаны N неотрицательных целых чисел — количество очков, набранных каждой командой. В третьей строке содержатся N чисел — разница забитых и пропущенных мячей каждой команды.

Формат выходных данных

Выведите N чисел — номера команд в порядке, в котором они будут записаны в итоговой таблице. Если количество очков и разница забитых и пропущенных мячей для каких-то команд совпадают, то их относительный порядок должен быть тем же, что и во входных данных.

Примеры

football.in	football.out
5 5 12 5 0 5 -2 8 2 -6 2	2 3 5 1 4

Задача G. Точки и отрезки

Имя входного файла: `segments.in`
Имя выходного файла: `segments.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дано n отрезков на числовой прямой и m точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b , если выполняется двойное неравенство $\min(a, b) \leq x \leq \max(a, b)$.

Формат входных данных

Первая строка содержит два целых числа n ($1 \leq n \leq 10^5$) — число отрезков и m ($1 \leq m \leq 10^5$) — число точек. В следующих n строках записаны по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке записаны m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходных данных

В выходной файл выведите m чисел — для каждой точки выведите количество отрезков, в которых она содержится.

Примеры

segments.in	segments.out
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1