

Задача А. Мутанты

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

Уже долгое время в Институте Искусств, Мутантов и Информационных Технологий разводят милых разноцветных зверюшек. Для удобства каждый цвет обозначен своим номером, всего цветов не более 10^9 . В один из прекрасных дней в питомнике случилось чудо: все зверюшки выстроились в ряд в порядке возрастания цветов. Пользуясь случаем, лаборанты решили посчитать, сколько зверюшек разных цветов живет в питомнике, и, по закону жанра, попросили вас написать программу, которая поможет им в решении этой нелегкой задачи.

Формат входного файла

В первой строке входного файла содержится единственное число N ($0 \leq N \leq 10^6$) — количество зверюшек в Институте. В следующей строке находятся N упорядоченных по убыванию неотрицательных целых чисел, не превосходящих 10^9 и разделенных пробелами — их цвета. В третьей строке файла записано число M ($1 \leq M \leq 200\,000$) — количество запросов вашей программе, в следующей строке через пробел записаны M целых неотрицательных чисел (не превышающих $10^9 + 1$).

Формат выходного файла

Выходной файл должен содержать M строчек. Для каждого запроса выведите число зверюшек заданного цвета в питомнике.

Примеры

<code>mutants.in</code>	<code>mutants.out</code>
10	1
1 1 3 3 5 7 9 18 18 57	2
5	1
57 3 9 1 179	2
	0

Задача В. Коровы - в стойла

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

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное расстояние между коровами было как можно больше.

Формат входного файла

В первой строке вводятся числа N ($2 < N < 10\,001$) — количество стойл и K

($1 < K < N$) — количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания — координаты стойл (координаты не превосходят 10^9).

Формат выходного файла

Выведите одно число — наибольшее возможное допустимое расстояние.

Примеры

<code>cows.in</code>	<code>cows.out</code>
5 3	99
1 2 3 100 1000	

Задача С. Ярый коллекционер бабочек

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

Как известно, Андрей Сергеевич — ярый коллекционер бабочек. Он имеет огромную коллекцию, экспонаты которой собраны со всего мира. Будем считать, что в мире существует 2 000 000 000 видов бабочек.

Чтобы не запутаться, Андрей Сергеевич присвоил каждому виду уникальный номер. Нумерация видов бабочек начинается с единицы.

Теперь он хочет знать, есть ли бабочка с видом K в его коллекции, или же её придётся добывать, затрачивая уйму сил и денег.

Формат входного файла

В первой строке входного файла содержится единственное число N ($1 \leq N \leq 100\,000$) — количество видов бабочек в коллекции Андрея Сергеевича.

В следующей строке через пробел находятся N упорядоченных по возрастанию чисел — номера видов бабочек в коллекции.

Все виды бабочки в коллекции имеют различные номера.

В третьей строке файла записано число M ($1 \leq M \leq 100\,000$), количество видов бабочек, про которых Андрей Сергеевич хочет узнать, есть ли они у него в коллекции или же нет. В последней строке входного файла содержатся через пробел M чисел — номера видов бабочек, наличие которых необходимо проверить.

Формат выходного файла

Выходной файл должен содержать M строчек. Для каждого запроса выведите число "YES", если бабочка с данным номером содержится в коллекции, и "NO" — в противном случае.

Примеры

collect.in	collect.out
7	NO
10 47 50 63 89 90 99	NO
4	YES
84 33 10 82	NO

Задача D. Оценка сортировки

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

Вам дан реализованный алгоритм сортировки, который требует $c * n * \lg(n)$ наносекунд, чтобы отсортировать n целых чисел. \lg здесь обозначает двоичный логарифм. Дано $time$ наносекунд, выведите в ответ наибольшее вещественное число n такое, что $c * n * \lg(n) \leq time$.

Формат входного файла

Входной файл содержит число c ($1 \leq c \leq 100$) и число $time$ ($1 \leq time \leq 2000000000$). Оба числа целые.

Формат выходного файла

В выходной файл должен быть содержать ответ. Ответ считается правильным, если он отличается от ответа жюри не более чем на 10^{-5}

Примеры

sortestimate.in	sortestimate.out
1 8	4.000000000

Note

$\lg(n) = \ln(n)/\ln(2)$, где \ln — натуральный логарифм (\log в языке Си, Ln в языке паскаль).

Задача E. Количество инверсий

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

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \dots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 50\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов

массива A .

Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

Примеры

inverse.in	inverse.out
4 1 2 4 5	0
4 5 4 2 1	6

Задача F. Точки и отрезки

Имя входного файла: `segments.in`
Имя выходного файла: `segments.out`
Ограничение по времени: 2 секунды
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 входного файла

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 по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходного файла

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

Примеры

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

Задача G. Min Max Search

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

Найдите минимальное число, которое входит в массив максимальное количество раз.

Формат входного файла

На первой строке дано целое число $1 \leq N \leq 10^6$, на второй строке записано ровно N чисел исходного массива $-10^9 \leq a_i \leq 10^9$.

Формат выходного файла

Выведите одно число - ответ на задачу.

Примеры

<code>minmaxsearch.in</code>	<code>minmaxsearch.out</code>
7 4 5 3 2 4 3 5	3

Задача H. Сортировка подсчетом

Имя входного файла: `countsort.in`
Имя выходного файла: `countsort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 4 мегабайта

Вам дан массив. Требуется его отсортировать.

Формат входного файла

В первой строке входного файла находится одно целое число n — количество элементов массива ($1 \leq n \leq 2000000$). Во второй строчке находятся n натуральных чисел — элементы массива. Все элементы массива не превосходят 10^4 .

Формат выходного файла

В единственную строку выходного файла выведите отсортированный массив.

Примеры

<code>countsort.in</code>	<code>countsort.out</code>
3 1 2 3	1 2 3
3 3 2 1	1 2 3

Note

Используйте силу, силу сортировки подсчетом.