

Задача А. Хип ли?

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

Структуру данных Heap можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Heap'a*, которое заключается в следующем. Для каждого $1 \leq i \leq n$ выполняются следующие условия:

- Если $2i \leq n$, то $a[i] \leq a[2i]$
- Если $2i + 1 \leq n$, то $a[i] \leq a[2i + 1]$

Дан массив целых чисел. Определите является ли он Heap'ом.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка содержит n целых чисел по модулю не превосходящих $2 \cdot 10^9$.

Формат выходного файла

Выведите «YES», если массив является Heap'ом и «NO» в противном случае.

Примеры

<code>isheap.in</code>	<code>isheap.out</code>
5 1 0 1 2 0	NO
5 1 3 2 5 4	YES

Задача В. Хипуй!

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

В этой задаче вам необходимо организовать структуру данных Heap для хранения целых чисел, над которой определены следующие операции:

- `Insert(X)` — добавить в Heap число X
- `Extract` — достать из Heap наибольшее число (удалив его при этом)

Формат входного файла

Во входном файле записано количество команд N ($1 \leq N \leq 100000$), потом последовательность из N команд, каждая в своей строке.

Каждая команда имеет такой формат: "0 <число>" или "1", обозначающие соответственно операции `Insert(<число>)` и `Extract`.

Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

Формат выходного файла

В выходной файл для каждой команды извлечения необходимо отдельно вывести число, полученное при выполнении команды `Extract`.

Примеры

<code>heap.in</code>	<code>heap.out</code>
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

Задача С. 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 ($2 < N < 10001$) — 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 D. 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 2000000000 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 ($1 \leq N \leq 100000$) — 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 ($1 \leq M \leq 100000$), 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 Е. 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 ($0 \leq N \leq 1000000$) — количество зверюшек в Институте. В следующей строке через пробел находятся N упо-

рядоченных по возрастанию чисел — их цвета. В третьей строке файла записано число M ($0 \leq M \leq 200000$) — 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

mutants.in	mutants.out
10	1
1 1 3 3 5 7 9 18 18 57	2
5	1
57 3 9 1 179	2
	0