

## Задача 4. lower bound

Источник:	основная*
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Функция `bsearch` из стандартной библиотеки языка C обладает большим недостатком: если в массиве нет того элемента, который мы ищем, то функция не возвращает абсолютно никакой информации. Это означает, что в некоторых случаях она бесполезна, например при решении задачи «Поиск ближайшего».

Гораздо полезнее функция `lower_bound` из стандартной библиотеки языка C++. Эта функция запускается для отсортированного массива  $A$  и элемента  $X$ , и возвращает номер первого элемента в массиве, который больше или равен  $X$ . Если такого элемента нет, то функция возвращает длину массива  $N$ . Иными словами, функция возвращает, сколько элементов в массиве  $A$  строго меньше заданного  $X$ .

В данной задаче вам предлагается реализовать `lower_bound` на языке C аналогично тому, как реализована функция `bsearch`. Это означает, что должны выполняться следующие требования:

1. Функция должна быть применима к массиву элементов любого типа. Это означает, что она должна принимать размер одного элемента в байтах и нетипизированные указатели, аналогично `qsort` и `bsearch`.
2. Функция должна поддерживать задание критерия сравнения для элементов. Это означает, что она должна принимать компаратор в виде указателя на функцию (можно без контекста).
3. Функция должна брать информацию только из своих параметров/аргументов. Иными словами, внутри неё нельзя обращаться к глобальным и статическим переменным.

Данную функцию нужно применить к двум заданным массивам: один состоит из целых чисел, а другой — из строк. Обратите внимание, что изначально массивы не отсортированы — вам следует применить к ним `qsort`.

### Формат входных данных

Входные данные задаются в таком порядке: сначала массив целых чисел, затем массив строк, затем запросы для массива целых чисел, и, наконец, запросы для массива строк.

В первом блоке записано одно целое число  $N_1$  — длина массива целых чисел ( $1 \leq N_1 \leq 10^5$ ). Далее записаны элементы этого массива:  $N_1$  целых чисел, каждое по абсолютной величине не превышает  $10^{15}$ .

Во втором блоке записано одно целое число  $N_2$  — длина массива строк ( $1 \leq N_2 \leq 10^5$ ). Далее записаны элементы этого массива:  $N_2$  непустых строк из маленьких букв латинского алфавита, длиной не более 31 символа каждая.

В третьем блоке записано целое число  $Q_1$  — количество запросов для массива целых чисел ( $1 \leq Q_1 \leq 10^5$ ). В остальных  $Q_1$  строках записаны целые числа, определяющие запросы на поиск, каждое число не превышает  $10^{15}$  по абсолютной величине.

В четвёртом блоке записано целое число  $Q_2$  — количество запросов для массива строк ( $1 \leq Q_2 \leq 10^5$ ). В остальных  $Q_2$  строках записаны строки-запросы, состоящие из маленьких букв латинского алфавита, длиной от 1 до 31 символа.

## Формат выходных данных

В выходные данные нужно вывести сначала результаты применения `lower_bound` для запросов на массиве целых чисел, а затем результаты применения для запросов на массиве строк. Первый блок результатов должен содержать  $Q_1$  целых чисел в диапазоне от 0 до  $N_1$ , каждое число в отдельной строке. Второй блок результатов должен содержать  $Q_2$  целых чисел в диапазоне от 0 до  $N_2$ , каждое число в отдельной строке.

## Пример

input.txt	output.txt
5	2
-1000000000	2
3000000000	4
5000000000	2
-1000000000	4
3000000000	4
5	5
a	0
hello	0
a	3
ba	5
a	
8	
3000000000	
2999999999	
3000000001	
0	
5000000000	
4000000000	
7000000000	
-7000000000	
3	
a	
b	
hi	

## Пояснение к примеру

Массив целых чисел в отсортированном виде выглядит как:  $-G$ ,  $-G$ ,  $3G$ ,  $3G$ ,  $5G$  (для краткости обозначим  $G = 10^9$ ). Первый запрос в точности равен  $3G$ , он впервые встречается под индексом 2 в массиве. Второй запрос чуть меньше  $3G$ , и в массиве отсутствует, так что нужно вернуть индекс первого элемента больше него, а это 2. Третий запрос чуть больше  $3G$ , и в массиве также отсутствует. В массиве всего 4 числа меньше запрашиваемого, так что ответ равен 4. Последние два запроса показывают, что нужно возвращать, когда запрашиваемое число больше или меньше всего массива.

Массив строк в отсортированном виде выглядит так: a, a, a, ba, hello. Для запроса a ответ равен 0, т.к. элементов меньше в массиве нет. Для запроса b ответ равен 3, т.к. все три строки a меньше него, а строка ba больше него. Запрос hi больше всего массива, так что ответ равен длине массива.