

Задача 5. Алгоритм расширенного бинарного поиска

Имя входных файлов: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничения по времени: $O(\log N)$

Ограничения по памяти: не использовать вспомогательные массивы

Реализовать функции расширенного алгоритма бинарного поиска:

1. «поиск слева», `bs_left` – поиск максимального индекса элемента массива, значение которого меньше искомого;
2. «поиск справа», `bs_right` – поиск минимального индекса элемента массива, значение которого больше искомого;

Пусть задан следующий упорядоченный по не убыванию массив *Array*:

Idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Key	-1	1	1	2	3	3	4	4	4	5	7	7	7	7	8	9	10

Тогда:

<code>bs_left(Array, 3) = 3</code> <code>bs_right(Array, 3) = 6</code>	<code>bs_left(Array, 7) = 9</code> <code>bs_right(Array, 7) = 14</code>
<code>bs_left(Array, 10) = 15</code> <code>bs_right(Array, 10) = 17</code>	<code>bs_left(Array, 6) = 9</code> <code>bs_right(Array, 6) = 10</code>
<code>bs_left(Array, -2) = -1</code> <code>bs_right(Array, -2) = 0</code>	<code>bs_left(Array, 99) = 16</code> <code>bs_right(Array, 99) = 17</code>

Формат входных данных

Первая строка содержит значение `key`, для которого требуется осуществить поиск.

Вторая строка содержит $2 < N < 10^9$ – количество элементов массива ключей.

Следующие N строк содержат значения элементов массива ключей.

Пример:

```
7
17
-1
1
1
2
3
3
4
4
4
5
7
7
7
7
8
9
10
```

Формат выходных данных

Первое число строки выходных данных должно быть результатом поиска слева значения `key` в массиве *Array*. Второе число – результатом поиска справа значения `key` в массиве *Array*.

Пример (для значений `key`, N , *Array* заданных выше):

```
9 14
```