

# Рекордное удаление

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2 секунды         |
| Ограничение по памяти:  | 256 мегабайт      |

Назовём *рекордом* элемент массива, который является максимумом на префиксе, заканчивающемся в этом элементе. Более формально, элемент  $a_i$  является рекордом в массиве  $a$ , если  $a_j < a_i$  для каждого  $j < i$ .

Рассмотрим следующий процесс на некотором массиве  $a$ :

- На первом этапе выберем в массиве  $a$  все элементы, которые являются рекордами, и удалим их из  $a$ . При этом порядок оставшихся элементов не меняется.
- На втором этапе выберем в получившемся массиве  $a$  все элементы, которые теперь являются рекордами, и удалим их из  $a$ .
- ...
- На  $m$ -м этапе выберем в  $a$  все элементы, которые стали рекордами после предыдущего шага, и удалим их из  $a$ . Будем повторять так до тех пор, пока массив  $a$  не станет пустым.

Вам дана перестановка  $p$  длины  $n$  и  $q$  запросов, описываемые двумя числами  $l$  и  $k$  ( $1 \leq l \leq n$ ,  $1 \leq k \leq 20$ ). Для такого запроса мы рассмотрим массив  $a = [p_l, p_{l+1}, p_{l+2}, \dots, p_n]$  и выполним ровно  $k$  этапов удаления (если массив  $a$  станет пустым раньше, чем за  $k$  этапов, то он останется пустым). Вам требуется посчитать количество элементов, удалённых из массива  $a$  за все эти этапы. Обратите внимание, что сам массив  $p$  никак не меняется (то есть удаления в одном запросе никак не влияют на другие запросы).

Напомним, что перестановкой длины  $n$  является массив, состоящий из  $n$  различных целых чисел от 1 до  $n$  в произвольном порядке. Например,  $[2, 3, 1, 5, 4]$  — перестановка, но  $[1, 2, 2]$  не перестановка (2 встречается в массиве дважды) и  $[1, 3, 4]$  тоже не перестановка ( $n = 3$ , но в массиве встречается 4).

## Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $1 \leq n, q \leq 10^5$ ) — длина перестановки и количество запросов.

Вторая строка содержит  $n$  различных целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ) — элементы перестановки  $p$ .

Каждая из следующих  $q$  строк содержит два целых числа  $l$  и  $k$  ( $1 \leq l \leq n$ ,  $1 \leq k \leq 20$ ) — описание очередного запроса.

## Формат выходных данных

Для каждого запроса выведите одно целое число — количество элементов, удалённых из соответствующего массива за все этапы.

## Система оценки

Тесты к этой задаче состоят из примеров и 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Группы в данной задаче «склеиваются». Это означает, что за подгруппу начисляются баллы, если хотя бы одна из ваших посылок успешно проходит все её тесты. Обратите внимание, что если у подгруппы имеются необходимые подгруппы, то для проверки на данной подгруппе текущая посылка должна пройти тесты во всех этих необходимых подгруппах (при этом результаты предыдущих посылок не учитываются).

| Группа | Баллы | Доп. ограничения |              |             | Необх. группы | Комментарий       |
|--------|-------|------------------|--------------|-------------|---------------|-------------------|
|        |       | $n$              | $q$          | $k$         |               |                   |
| 0      | 0     | —                | —            | —           | —             | Тесты из условия  |
| 1      | 17    | $n \leq 100$     | $q \leq 100$ | $k \leq 10$ | —             | —                 |
| 2      | 12    | —                | —            | —           | —             | $p_i = n - i + 1$ |
| 3      | 22    | —                | —            | $k = 1$     | —             | —                 |
| 4      | 21    | —                | —            | $k \leq 2$  | 3             | —                 |
| 5      | 28    | —                | —            | —           | 0 – 4         | —                 |

## Примеры

| стандартный ввод  | стандартный вывод          |
|---|----------------------------|
| 3 2<br>2 3 1<br>2 2<br>1 1                                      | 2<br>2                     |
| 4 3<br>4 3 2 1<br>4 1<br>3 3<br>1 3                             | 1<br>2<br>3                |
| 7 6<br>3 5 1 4 6 2 7<br>1 1<br>1 2<br>1 3<br>3 3<br>3 1<br>6 20 | 4<br>6<br>7<br>5<br>4<br>2 |

## Замечание

Рассмотрим первый пример:

- В первом запросе  $l = 2$  и  $k = 2$ ,  $a = [3, 1]$ . На первом этапе удалится элемент 3, после этого  $a = [1]$ . На втором этапе удалится элемент 1, после этого  $a = []$ . Таким образом, за 2 этапа удалилось 2 элемента.
- Во втором запросе  $l = 1$  и  $k = 1$ ,  $a = [2, 3, 1]$ . На первом этапе удалятся элементы 2 и 3, после этого  $a = [1]$ . Таким образом, за 1 этап удалилось 2 элемента.

Рассмотрим второй пример:

- В первом запросе  $l = 4$  и  $k = 1$ ,  $a = [1]$ . На первом этапе удалится элемент 1, после этого  $a = []$ . Таким образом, за 1 этап удалился 1 элемент.
- Во втором запросе  $l = 3$  и  $k = 3$ ,  $a = [2, 1]$ . На первом этапе удалится элемент 2, после этого  $a = [1]$ . На втором этапе удалится элемент 1, после этого  $a = []$ . После третьего этапа массив  $a$  останется пустым. Таким образом, за 3 этапа удалилось 2 элемента.
- В третьем запросе  $l = 1$  и  $k = 3$ ,  $a = [4, 3, 2, 1]$ . На первом этапе удалится элемент 4, после этого  $a = [3, 2, 1]$ . На втором этапе удалится элемент 3, после этого  $a = [2, 1]$ . На третьем этапе удалится элемент 2, после этого  $a = [1]$ . Таким образом, за 3 этапа удалилось 3 элемента.