

[ГЛАВНАЯ](#) [ТОП](#) [КАТАЛОГ](#) [СОРЕВНОВАНИЯ](#) [ТРЕНИРОВКИ](#) [АРХИВ](#) [ГРУППЫ](#) [РЕЙТИНГ](#) [EDU](#) [API](#) [КАЛЕНДАРЬ](#) [ПОМОЩЬ](#)
ТЕХНОКУБОК 
[ЗАДАЧИ](#) [ОТΟΣЛАТЬ](#) [МОИ ПОСЫЛКИ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

## С. Астроград

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В Астрополисе прошел концерт популярной группы Астроград. За пару дней до концерта перед кассой выстроилась огромная очередь из людей, желающих туда попасть. Изначально очередь была пуста. В каждый из  $n$  моментов времени происходило следующее:

1. В очередь пришел новый человек с уникальным номером  $id$ , он встает в очередь последним.
2. Человеку, стоящему спереди очереди, удалось купить билет. Он уходит.
3. Человеку, стоящему последнему в очереди, надоело ждать. Он уходит.
4. Человек с уникальным номером  $q$  хочет знать, сколько людей стоит в очереди спереди него.
5. Очередь хочет знать, человек с каким уникальным номером стоит сейчас первым и задерживает всех.

Вам необходимо написать программу, которая умеет обрабатывать описанные события.

### Входные данные

В первой строке дано целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество событий. В каждой из следующих  $n$  строк дано описание событий: номер события, а также число  $id$  ( $1 \leq id \leq 10^5$ ) для событий типа 1 и число  $q$  для событий типа 4. События происходили в том порядке, в каком они описаны во входном файле. Гарантируется корректность всех событий.

### Выходные данные

Выведите ответы для событий типа 4 и 5 в том порядке, в каком они описаны во входном файле.

### Пример

входные данные	Скопировать
<pre>7 1 1 5 1 3 3 2 1 2 4 2</pre>	
выходные данные	Скопировать
<pre>1 0</pre>	

### Примечание

В примере из условия происходили следующие события:

1. В очередь пришел человек с  $id = 1$ . Очередь: [ 1 ]
2. Первым в очереди стоит человек с  $id = 1$ . Очередь: [ 1 ]
3. В очередь пришел человек с  $id = 3$ . Очередь: [ 1, 3 ]

### Университет ИТМО. Алгоритмы и структуры данных y2019

Открытая

Участник



### y2019-1-2. Стеки, очереди, СНМ

Закончено

Дорешивание



### → Виртуальное участие

Виртуальное соревнование – это способ прорешать прошедшее соревнование в режиме, максимально близком к участию во время его проведения. Поддерживается только ICPC режим для виртуальных соревнований. Если вы раньше видели эти задачи, виртуальное соревнование не для вас – решайте эти задачи в архиве. Если вы хотите просто дорешать задачи, виртуальное соревнование не для вас – решайте эти задачи в архиве. Запрещается использовать чужой код, читать разборы задач и общаться по содержанию соревнования с кем-либо.

[Начать виртуальное участие](#)

### → Отослать?

Язык: GNU G++17 7.3.0

Выберите файл: Выберите файл Файл ...выбран

[Отослать](#)

### → Последние послылки

Посылка	Время	Вердикт
<a href="#">64530749</a>	08.11.2019 18:21	Полное решение
<a href="#">64530573</a>	08.11.2019 18:17	Неправильный ответ на тесте 1

4. Последнему в очереди надоело стоять и он уходит. Очередь: [ 1 ]
5. Первому в очереди удалось купить билет и он уходит. Очередь: [ ]
6. В очередь пришел человек с  $id = 2$ . Очередь: [ 2 ]
7.  $q = 2$  хочет знать, сколько человек стоит перед ним. Очередь: [ 2 ]

---

[Codeforces](#) (c) Copyright 2010-2022 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 18.03.2022 20:52:42 (f1).  
Десктопная версия, переключиться на [мобильную](#).  
[Privacy Policy](#)

При поддержке



УНИВЕРСИТЕТ ИТМО