# Задание по программированию: Система бронирования отелей

## Условие

Разработайте систему бронирования отелей, позволяющую бронировать номера клиентами и контролировать спрос владельцами отелей. Ваша программа должна обрабатывать следующие события:

* **BOOK time hotel\_name client\_id room\_count** — забронировать клиентом ***client\_id room\_count*** номеров в отеле ***hotel\_name*** в момент времени ***time***. Время измеряется в секундах, отсчитываемых от некоторого момента.
* **CLIENTS hotel\_name** — вывести количество различных клиентов, бронировавших номера в отеле ***hotel\_name*** за последние сутки. Более формально интересующий диапазон времени описывается следующим образом: пусть current\_time — время последнего события **BOOK**, тогда в данном запросе нужно учесть все события с current\_time − 86400 < ***time*** ⩽ current\_time, где 86400 — количество секунд в сутках. Обратите внимание, что последнее бронирование должно учитываться, а бронирование, случившееся ровно за сутки до него, учитываться не должно.
* **ROOMS hotel\_name** — вывести количество номеров, забронированных за последние сутки в отеле ***hotel\_name***.

## Формат входных данных

В первой строке вводится количество запросов Q — натуральное число, не превосходящее 10^5. В следующих Q строках в соответствии с описанным выше форматом вводятся запросы. Гарантируется, что:

* ***time*** — целое число в диапазоне от −10^18 до 10^18 и не убывает от события к событию.
* ***hotel\_name*** — строка из латинских букв и цифр, имеющая длину не более 12 символов.
* ***client\_id*** — натуральное число, не превосходящее 10^9.
* ***room\_count*** — целое положительное число, не превосходящее 1000.

## Формат выходных данных

Для каждого запроса **CLIENTS** и **ROOMS** выведите единственное целое число — ответ на запрос. Если указанный в запросе отель пока не имеет ни одного бронирования, выведите 0.

## Ограничения

1 секунда на выполнение всех запросов. Все описанные в условии гарантии действительно справедливы для всех тестов, на которых будет запускаться ваша программа. Проверять корректность тестов не нужно.

## Пример

### Ввод

11

CLIENTS Marriott

ROOMS Marriott

BOOK 10 FourSeasons 1 2

BOOK 10 Marriott 1 1

BOOK 86409 FourSeasons 2 1

CLIENTS FourSeasons

ROOMS FourSeasons

CLIENTS Marriott

BOOK 86410 Marriott 2 10

ROOMS FourSeasons

ROOMS Marriott

### Вывод

0

0

2

3

1

1

10

### Комментарии к примеру

После бронирования, случившегося в момент времени 86410, в статистике перестают учитываться бронирования, случившиеся в момент времени 10.

## Подсказки

Если задача вызывает непреодолимые трудности, вы можете воспользоваться подсказками. Они помогут быстрее сдать задачу, но в итоге вы получите меньше пользы от процесса поиска решения.

Не нужно открывать все подсказки сразу. Используйте их последовательно.

## Подсказка 1

Удобно представить себе систему бронирования отелей как сервис, непрерывно обрабатывающий поток запросов на бронирование и иногда возвращающий какую-то статистику. Вооружившись этой мыслью, ответьте на два вопроса: • Есть ли необходимость хранить старые события, или, обработав событие, о нём можно забыть? • Если да, то нужно ли хранить вообще все старые события, или с какого-то момента они-таки становятся неинтересными?

## Подсказка 2

Система бронирования отелей, подсказка 2 Старые события нужно хранить, потому что каждое из них пригождается ещё раз — через сутки после появления. С другой стороны, спустя сутки события уже не нужны, и их можно удалить. Какая структура данных подходит для эффективного добавления новых событий и удаления старых?

## Подсказка 3

Разобравшись со структурой для хранения событий, зададимся вопросом о запросах CLIENTS и ROOMS. Поскольку они интересуют нас поотельно, словари с названием отеля в ключе и какими-то значениями напрашиваются сами собой. Но можно ли придумать такие значения для этих словарей, чтобы оба запроса на статистику заключались в нескольких элементарных операциях? Иными словами, можно ли хранить и поддерживать практически готовые ответы на запросы CLIENTS и ROOMS для каждого отеля?

## Подсказка 4

С запросом ROOMS всё просто: будем хранить для каждого отеля количество забронированных номеров. Когда случилось новое бронирование в данном отеле, увеличим это количество на нужную величину. Когда пришла пора удалять старое бронирование, уменьшим.

## Подсказка 5

С запросом CLIENTS всё обстоит чуть интереснее, но укладывается в ту же самую концепцию. Вам нужен объект или структура данных, которая позволит делать следующее: • обработать новое бронирование от конкретного клиента; • удалить старое бронирование от конкретного клиента; • вернуть текущее количество клиентов, имеющих хотя бы одно бронирование.