

5 Хеш-таблицы

5.1 Телефонная книга

Телефонная книга

Реализовать структуру данных, эффективно обрабатывающую запросы вида `add number name`, `del number` и `find number`.

Вход. Последовательность запросов вида `add number name`, `del number` и `find number`, где `number` — телефонный номер, содержащий не более семи знаков, а `name` — короткая строка.

Выход. Для каждого запроса `find number` выведите соответствующее имя или сообщите, что такой записи нет.

Цель в данной задаче — реализовать простую телефонную книгу, поддерживающую три следующих типа запросов. С указанными ограничениями данная задача может быть решена с использованием таблицы с прямой адресацией.



- `add number name`: добавить запись с именем `name` и телефонным номером `number`. Если запись с таким телефонным номером уже есть, нужно заменить в ней имя на `name`.
- `del number`: удалить запись с соответствующим телефонным номером. Если такой записи нет, ничего не делать.
- `find number`: найти имя записи с телефонным номером `number`. Если запись с таким номером есть, вывести имя. В противном случае вывести «not found» (без кавычек).

Формат входа. Первая строка содержит число запросов n . Каждая из следующих n строк задаёт запрос в одном из трёх описанных выше форматов.

Формат выхода. Для каждого запроса `find` выведите в отдельной строке либо имя, либо «not found».

Ограничения. $1 \leq n \leq 10^5$. Телефонные номера содержат не более семи цифр и не содержат ведущих нулей. Имена содержат только буквы латинского алфавита, не являются пустыми строками и имеют длину не больше 15. Гарантируется, что среди имён не встречается строка «not found».

Пример.

Вход:

```
12
add 911 police
add 76213 Mom
add 17239 Bob
find 76213
find 910
find 911
del 910
del 911
find 911
find 76213
add 76213 daddy
find 76213
```

Выход:

```
Mom
not found
police
not found
Mom
daddy
```

Пример.

Вход:

```
8
find 3839442
add 123456 me
add 0 granny
find 0
find 123456
del 0
del 0
find 0
```

Выход:

```
not found
granny
me
not found
```