

## Задача А. Set (!) (2 балла)

Имя входного файла: `set.in`  
Имя выходного файла: `set.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Реализуйте множество с использованием хеш таблицы. Использовать стандартную библиотеку (`set`, `map`, `LinkedHashMap`, и т. п.) не разрешается.

### Формат входного файла

Входной файл содержит описание операций, их количество не превышает 1000000. В каждой строке находится одна из следующих операций:

- **insert**  $x$  — добавить элемент  $x$  в множество. Если элемент уже есть в множестве, то ничего делать не надо.
- **delete**  $x$  — удалить элемент  $x$ . Если элемента  $x$  нет, то ничего делать не надо.
- **exists**  $x$  — если ключ  $x$  есть в множестве выведите «true», если нет «false».

В множество помещаются и извлекаются только целые числа, не превышающие по модулю  $10^9$ .

### Формат выходного файла

Выведите последовательно результат выполнения всех операций **exists**. Следуйте формату выходного файла из примера.

### Пример

| set.in   | set.out |
|----------|---------|
| insert 2 | true    |
| insert 5 | false   |
| insert 3 | false   |
| exists 2 |         |
| exists 4 |         |
| insert 2 |         |
| delete 2 |         |
| exists 2 |         |

## Задача В. Мар (2 балла)

Имя входного файла: `map.in`  
Имя выходного файла: `map.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Реализуйте ассоциативный массив с использованием хеш таблицы. Использовать стандартную библиотеку (`set`, `map`, `LinkedHashMap`, и т. п.) не разрешается.

### Формат входного файла

Входной файл содержит описание операций, их количество не превышает 100000. В каждой строке находится одна из следующих операций:

- `put x y` — поставить в соответствие ключу `x` значение `y`. Если ключ уже есть, то значение необходимо изменить.
- `delete x` — удалить ключ `x`. Если элемента `x` нет, то ничего делать не надо.
- `get x` — если ключ `x` есть в ассоциативном массиве, то выведите соответствующее ему значение, иначе выведите «none».

Ключи и значения — строки из латинских букв длиной не более 20 символов.

### Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`. Следуйте формату выходного файла из примера.

### Пример

| map.in           | map.out |
|------------------|---------|
| put hello privet | privet  |
| put bye poka     | poka    |
| get hello        | none    |
| get bye          |         |
| delete hello     |         |
| get hello        |         |

## Задача С. LinkedHashMap (2 балла)

Имя входного файла: `linkedmap.in`  
Имя выходного файла: `linkedmap.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Реализуйте прошитый ассоциативный массив с использованием хеш таблицы. Использовать стандартную библиотеку (`set`, `map`, `LinkedHashMap`, и т. п.) не разрешается.

### Формат входного файла

Входной файл содержит описание операций, их количество не превышает 100000. В каждой строке находится одна из следующих операций:

- `put x y` — поставить в соответствие ключу `x` значение `y`. Если элемент уже есть, то значение необходимо изменить.
- `delete x` — удалить ключ `x`. Если элемента `x` нет, то ничего делать не надо.
- `get x` — если ключ `x` есть в множестве выведите соответствующее ему значение, если нет выведите «none».
- `prev x` — вывести значение соответствующее ключу находящемуся в ассоциативном массиве, который был вставлен позже всех, но до `x` или «none», если такого нет или в массиве нет `x`.
- `next x` — вывести значение соответствующее ключу находящемуся в ассоциативном массиве, который был вставлен раньше всех, но после `x` или «none», если такого нет или в массиве нет `x`.

Ключи и значения — строки из латинских букв длиной не более 20 символов.

### Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`, `prev`, `next`. Следуйте формату выходного файла из примера.

### Пример

| linkedmap.in | linkedmap.out |
|--------------|---------------|
| put zero a   | c             |
| put one b    | b             |
| put two c    | d             |
| put three d  | c             |
| put four e   | a             |
| get two      | e             |
| prev two     | none          |
| next two     |               |
| delete one   |               |
| delete three |               |
| get two      |               |
| prev two     |               |
| next two     |               |
| next four    |               |

## Задача D. MultiMap (2 балла)

Имя входного файла: `multimap.in`  
Имя выходного файла: `multimap.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Реализуйте множественное отображение с использованием хеш таблиц. Использовать стандартную библиотеку (`set`, `map`, `LinkedHashMap`, и т. п.) не разрешается.

### Формат входного файла

Входной файл содержит описание операций, их количество не превышает 100000. В каждой строке находится одна из следующих операций:

- `put x y` — добавить пару  $(x, y)$ . Если пара уже есть, то второй раз её добавлять не надо.
- `delete x y` — удалить пару  $(x, y)$ . Если пары нет, то ничего делать не надо.
- `deleteall x` — удалить все пары с первым элементом  $x$ .
- `get x` — вывести количество пар с первым элементом  $x$ , а затем вторые элементы всех этих пар в произвольном порядке.

Ключи и значения — строки из латинских букв длиной не более 20 символов.

### Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`. Следуйте формату выходного файла из примера. Гарантируется, что размер выходного файла не превысит 10 мегабайт.

### Пример

| multimap.in              | multimap.out         |
|--------------------------|----------------------|
| <code>put a a</code>     | <code>3 b c a</code> |
| <code>put a b</code>     | <code>2 c a</code>   |
| <code>put a c</code>     | <code>0</code>       |
| <code>get a</code>       |                      |
| <code>delete a b</code>  |                      |
| <code>get a</code>       |                      |
| <code>deleteall a</code> |                      |
| <code>get a</code>       |                      |