

## 5.2 Хеширование цепочками

Хеширование цепочками — один из наиболее популярных методов реализации хеш-таблиц на практике. Ваша цель в данной задаче — реализовать такую схему, используя таблицу с  $m$  ячейками и полиномиальной хеш-функцией на строках

$$h(S) = \left( \sum_{i=0}^{|S|-1} S[i]x^i \bmod p \right) \bmod m,$$

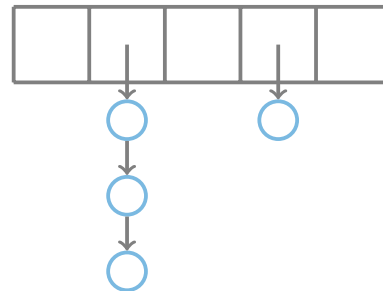
где  $S[i]$  — ASCII-код  $i$ -го символа строки  $S$ ,  $p = 1\,000\,000\,007$  — простое число, а  $x = 263$ . Ваша программа должна поддерживать следующие типы запросов:

- `add string`: добавить строку `string` в таблицу. Если такая строка уже есть, проигнорировать запрос;
- `del string`: удалить строку `string` из таблицы. Если такой строки нет, проигнорировать запрос;
- `find string`: вывести «yes» или «no» в зависимости от того, есть в таблице строка `string` или нет;
- `check i`: вывести  $i$ -й список (используя пробел в качестве разделителя); если  $i$ -й список пуст, вывести пустую строку.

При добавлении строки в цепочку, строка должна добавляться **в начало** цепочки.

**Формат входа.** Первая строка размер хеш-таблицы  $m$ . Следующая строка содержит количество запросов  $n$ . Каждая из последующих  $n$  строк содержит запрос одного из перечисленных выше четырёх типов.

**Формат выхода.** Для каждого из запросов типа `find` и `check` выведите результат в отдельной строке.



**Ограничения.**  $1 \leq n \leq 10^5$ ;  $\frac{n}{5} \leq m \leq n$ . Все строки имеют длину от одного до пятнадцати и содержат только буквы латинского алфавита.

**Пример.**

Вход:

```
5
12
add world
add Hello
check 4
find World
find world
del world
check 4
del Hello
add luck
add Good
check 2
del good
```

Выход:

```
Hello world
no
yes
Hello
Good luck
```

ASCII коды букв 'w', 'o', 'r', 'l', 'd' равны 119, 111, 114, 108, 100, соответственно. Поэтому

$$h(\text{world}) = (119 + 111 \times 263 + 114 \times 263^2 + 108 \times 263^3 + 100 \times 263^4 \bmod 1\,000\,000\,007) \bmod 5 = 4.$$

Оказывается, что  $h(\text{Hello})$  тоже равно четырём. Поскольку новые строки добавляются в начало списка, после второго запроса add список содержит строки Hello и world (именно в таком порядке). Строка World не находится, а world находится. После удаления строки world в цепочке 4 остаётся только строка Hello. И так далее.

**Пример.**

Вход:

```
4
8
add test
add test
find test
del test
find test
find Test
add Test
find Test
```

Выход:

```
yes
no
no
yes
```

**Пример.**

Вход:

```
3
12
check 0
find help
add help
add del
add add
find add
find del
del del
find del
check 0
check 1
check 2
```

Выход:

```
no
yes
yes
no

add help
```

Обратите внимание на то, что нужно выводить пустую строку в случае, если соответствующая цепочка пуста. Строки в запросах могут совпадать с названиями запросов.

**Указания.**

- Будьте осторожны с переполнением целого типа. Используйте `long long` в C++ и `long` в Java при необходимости. При вычислении значения многочлена по модулю  $p$  берите результат по модулю  $p$  после каждой арифметической операции.
- Будьте осторожны с отрицательными числами по модулю  $p$ . Во многих языках программирования  $(-2)\%5 \neq 3\%5$ . Один

из способов избежать этого — использовать  $x \leftarrow ((a \% p) + p) \% p$  вместо  $x \leftarrow a \% p$ .