

Вариант №202

Написать программу для работы с таблицей, использующей два пространства ключей, по запросам оператора.

Каждый элемент таблицы имеет следующую структуру:

```
struct Item {
    // указатель на информацию
    InfoType *info;
    // версия элемента
    RelType release;
    // указатель на следующий элемент с данным составным ключом
    Item *next;

    // опциональные поля, для оптимизации выполнения операций,
    // состав и наличие которых должны быть обоснованы:

    // ключ элемента из 1-го пространства ключей
    KeyType1 key1;
    // ключ элемента из 2-го пространства ключей
    KeyType2 key2;
    // связь с элементом 1-го пространства ключей по индексу
    IndexType1 ind1;
    // связь с элементом 2-го пространства ключей по индексу
    IndexType2 ind2;
    // связь с элементом 2-го пространства ключей по указателю
    PointerType1 *p1;
    // связь с элементом 2-го пространства ключей по указателю
    PointerType2 *p2;
};
```

В таблице могут присутствовать элементы с одинаковыми составными ключами (key1, key2) и разными номерами версий (номер версии элемента формируется как порядковый номер элемента в последовательности элементов с одинаковыми ключами, определяемый при включении элемента в таблицу).

Первое пространство ключей организовано как просматриваемая таблица, организованная вектором; каждый элемент таблицы имеет следующую структуру:

```
struct KeySpace1 {
    // признак занятости элемента
    BusyType1 busy;
    // ключ элемента
    KeyType1 key;
    // указатель на информацию
    Item *info;
};
```

Максимальный размер пространства ключей ограничен величиной `msize1`, значение которой определяется при инициализации таблицы.

В пространстве не может быть двух элементов с одинаковыми ключами.

В данном пространстве ключей предусмотрены следующие особые операции:

- удаление из таблицы элементов, заданных диапазоном ключей, без реорганизации таблицы; в таблице могут отсутствовать элементы с ключами, задающими диапазон;
- реорганизация таблицы (сборка мусора: сжатие таблицы путем удаления свободных элементов).

Второе пространство ключей организовано как перемешанная таблица, использующая перемешивание сложением. Перемешанная таблица организована вектором; каждый элемент таблицы имеет следующую структуру:

```
struct KeySpace2 {  
    // признак занятости элемента  
    BusyType2 busy;  
    // ключ элемента  
    KeyType2 key;  
    // указатель на информацию  
    InfoType *info;  
};
```

Максимальный размер пространства ключей ограничен величиной `msize2`, значение которой определяется при инициализации таблицы.

В пространстве не может быть двух элементов с одинаковыми ключами.

В данном пространстве ключей особые операции не предусмотрены.