

Вариант №144

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;

    // 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 не могут присутствовать два элемента с одинаковыми составными ключами (key1, key2).

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

```
struct KeySpace1 {
    // ненулевой ключ элемента
    KeyType1 key;
    // ключ родительского элемента, может быть нулевым
    KeyType1 par;
    // указатель на информацию
    Item *info;
    // указатель на следующий элемент
    KeySpace1 *next;
};
```

В пространстве не может быть двух элементов с одинаковыми ключами, а значение ключа родительского элемента для элемента должно совпадать с каким-либо значением ключа элемента, существующего в таблице, либо быть нулевым.

В данном пространстве ключей предусмотрены следующие особые операции:

- удаление из таблицы элемента, заданного своим ключом; при этом также должны быть удалены (рекурсивно) все элементы, ссылающиеся на удаляемый;
- поиск в таблице всех элементов, заданных значением ключа родительского элемента; результатом поиска должна быть новая таблица, содержащая найденные элементы.

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;
// номер версии элемента
RelType2 release;
// указатель на информацию
InfoType *info;
};

```

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

В пространстве могут находиться несколько элементов с одинаковыми ключами и разными номерами версий (номер версии элемента формируется как порядковый номер элемента в последовательности элементов с одинаковыми ключами, определяемый при включении элемента в таблицу).

В данном пространстве ключей предусмотрены следующие особые операции:

- поиск в таблице всех версий элемента, заданного ключом, или конкретной (заданной) версии элемента, также заданного своим ключом; результатом поиска должна быть новая таблица, содержащая найденные элементы.