

### **Вариант 10\*\***

Написать программу для работы с просматриваемой таблицей по запросам оператора.

Просматриваемая таблица организована вектором; каждый элемент таблицы имеет следующую структуру:

```
struct Item{  
    int busy;           /* признак занятости элемента */  
    int key;            /* ключ элемента */  
    int release;        /* номер версии элемента */  
    char *info;          /* указатель на информацию */  
};
```

Максимальный размер таблицы ограничен (для задания максимального размера таблицы использовать константу – например, `const int SIZE = ...;`).

Предусмотреть следующие операции:

- включение нового элемента в таблицу при условии, что в таблице 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);
- удаление из таблицы всех версий элемента, заданного ключом, или конкретной (заданной) версии элемента, также заданного своим ключом, без реорганизации таблицы;
- 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;
- вывод содержимого таблицы на экран.

Разработать два варианта программы:

- a) и сама таблица, и информация, относящаяся к элементу таблицы, хранятся в основной памяти;
- b) и сама таблица, и информация, относящаяся к элементу таблицы, хранятся во внешней памяти (используется двоичный файл произвольного доступа). Все операции выполняются с таблицей, размещенной в основной памяти. Таблица считывается из файла (или создается в первый раз) в начале сеанса работы и записывается в файл в конце сеанса работы. Информация, относящаяся к элементу таблицы, записывается в файл сразу же при выполнении операции включения в таблицу. Имя файла вводится по запросу из программы.

#### **Примечания:**

1. Программа должна содержать несколько функций; функция `main` должна выполнять: вывод меню, ввод и анализ ответа, вызов на исполнение требуемой функции;
2. В программе нужно предусмотреть проверку правильности ввода данных;
3. Для варианта b) следует модифицировать структуру, определяющую элемент таблицы, включив в нее длину информации и ее смещение в файле;
4. В варианте b) для работы с файлом использовать функции пакета `stdio.h`; чтение и запись выполнять с помощью `fread()` и `fwrite()`, в которых должна быть указана реальная длина информации.