

# Лабораторная работа № 3б «Работа с хеш-таблицами»

## Введение

Необходимо спроектировать и разработать на языке C:

1. Прикладную программу, позволяющую пользователю в диалоговом режиме работать с таблицей.
2. Библиотеку, предоставляющую функциональность по работе с таблицей, размещенной в основной памяти (оперативной памяти).

## Структура таблицы

Таблица задаётся структурой:

```
struct Table {  
    // указатель на пространство ключей;  
    // для таблиц, использующих перемешивание сцеплением — Keyspace **ks  
    KeySpace *ks;  
  
    // опциональное поле, ограничивающее размер пространства ключей,  
    // его наличие определяется типом организации соответствующего пространства,  
    // в соответствии с условиями индивидуального задания  
  
    // размер области пространства ключей  
    IndexType msize;  
  
    // опциональное поле с текущим количеством элементов  
    // в пространстве ключей,  
    // его наличие определяется типом организации соответствующего пространства,  
    // в соответствии с условиями индивидуального задания  
  
    // количество элементов в области пространства ключей  
    IndexType csize;  
};
```

## Операции, поддерживаемые таблицей

Должны быть предусмотрены следующие операции:

1. включение нового элемента в таблицу с соблюдением ограничений на уникальность значений ключевой информации;
2. удаление из таблицы элемента по заданному значению ключа;
3. поиск в таблице элемента по заданному значению ключа, результатом поиска должны быть копии всех найденных элементов со значениями ключей;
4. вывод содержимого таблицы в поток;

5. импорт текущего состояния таблицы из бинарного файла;
6. экспорт текущего состояния таблицы в бинарный файл;
7. особые операции, в соответствии с индивидуальным заданием.

### **Дополнительное задание**

Реализовать автоматическое расширение таблицы при её полном заполнении (т. е. при достижении коэффициентом заполнения таблицы величины 1). При этом новый размер таблицы должен являться простым числом и выбираться автоматически.

### **Примечания**

1. Логически законченные части алгоритма решения задачи должны быть оформлены в виде отдельных функций с параметрами. Использование глобальных переменных не допускается.
2. Функции для работы с таблицами не должны быть диалоговыми, т. е. они должны принимать все необходимые данные в качестве параметров и возвращать результат работы в виде соответствующих структур данных и кодов ошибок (исключение: функции вывода таблицы в стандартный поток вывода или записи файл).
3. Диалоговые функции должны использовать описанные выше функции.
4. Программа должна осуществлять проверку корректности вводимых данных.
5. В случае возникновения ошибочных ситуаций при выполнении операций с таблицами программа должна выводить соответствующие сообщения, после чего продолжать работу.
6. Для сборки программы и библиотек должна использоваться система сборки (например: Make или CMake).
7. Библиотеки и прикладная программа должны собираться независимо друг от друга.
8. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: valgrind, санитайзеры, встроенные в IDE средства и т. д.