**Лабораторная №1 Записи. Реализация стека, очереди на статических структурах.**

**Вариант 1**

1. Создать запись Страна следующей структуры: Название, столица, количество населения. Написать программу, реализующую следующее меню:

* добавить элемент в файл
* удалить элемент из файла
* принадлежность элемента файлу
* вывод всех записей на экран

1. Создать стек целых чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
2. Создать простую очередь строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 2**

1. Создать запись Студент следующей структуры: ФИО, Дата\_рождения – строки, курс, группа- байтовые числа. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек вещественных чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь вещественных чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 3**

1. Создать запись Преподаватель следующей структуры: ФИО, Предмет, Ученая\_степень, Ученое\_звание– строки, стаж работы – байтовое число. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 4**

1. Создать запись Работник следующей структуры: ФИО, Должность, Образование– строки, стаж работы – байтовое число. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь вещественных чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 5**

1. Создать запись Город следующей структуры: Название– строка, численность населения, площадь – целые числа. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек целых чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 6**

1. Создать запись Школа следующей структуры: Название, год\_открытия – строка, численность учащихся, численность преподавателей – целые числа. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь вещественных чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 7**

1. Создать запись Дом следующей структуры: Адрес , год\_сдачи – строка, количество этажей, подъездов, квартир – целые числа. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек целых чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь вещественных чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 8**

1. Создать запись Абитуриент структуры: Школа, год окончания, Вуз поступления – строки, средний бал аттестата – целое число. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек целых строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь целых чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 9**

1. Создать запись Выпускник\_вуза: ФИО, Наименование\_Вуза, факультет, специальность – строки, возраст(количество полных лет) – целое число. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек записи Выпускник вуза на основе статического массива типа запись. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь строк на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Вариант 10**

1. Создать запись Спортсмен: ФИО, Вид\_спорта, факультет, специальность – строки, возраст(количество полных лет) – целое число. Написать программу, реализующую следующее меню:
   * добавить элемент в файл
   * удалить элемент из файла
   * принадлежность элемента файлу
   * вывод всех записей на экран
2. Создать стек записи Спортсмен на основе статического массива типа запись. Реализовать методы : Добавить элемент, удалить элемент, вершина стека.
3. Создать простую очередь вещественных чисел на основе статического массива. Реализовать методы : Добавить элемент, удалить элемент, длина очереди.

**Лабораторная №2. Сортировка. Поиск.**

**Вариант 1**

1. Сортировать массив целых чисел методом простых вставок, обменной сортировкой.. Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов массива не менее 10000.
2. Для сортированного массива A(1000) целых чисел реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарный поиск для массива записей следующей структуры: Фамилия, число полных лет. Методы реализовать для поля Фамилия.

**Вариант 2**

1. Сортировать массив целых чисел карманной сортировкой, быстрой сортировкой. Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов массива не менее 10000.
2. Для сортированного массива В(1000) вещественных чисел реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива записей следующей структуры: Фамилия, дата\_рождения, адрес. Методы реализовать для поля Фамилия.

**Вариант 3**

1. Сортировать файл целых чисел обменной сортировкой, сортировкой вставками. Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов файла не менее 10000.
2. Для сортированного массива В(1000) вещественных чисел реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива записей следующей структуры: Фамилия, дата\_рождения, адрес, Вуз(где учится). Методы реализовать для поля Фамилия.

**Вариант 4**

1. Сортировать файл структуры: Фамилия, дата\_рождения, адрес обменной сортировкой, сортировкой вставками по полю Фамилия. Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов файла не менее 10000.
2. Для сортированного массива Х(1000) вещественных чисел реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива записей следующей структуры: Фамилия, дата\_рождения, адрес. Методы реализовать для поля Фамилия.

**Вариант 5**

1. Сортировать массив обменной, быстрой, карманной сортировкой. Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов массива не менее 10000.
2. Для сортированного массива строк (фрукты) fr(1000) реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать метод последовательного поиска для массива записей следующей структуры: Фамилия, адрес. Метод реализовать для поля Фамилия.

**Вариант 6**

1. Сортировать массив обменной, быстрой, карманной сортировкой, сортировкой вставками . Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов массива не менее 10000.
2. Для сортированного массива строк (название книг) fr(1000) реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива записей следующей структуры: Фамилия, адрес, дата рождения. Методы реализовать для поля Дата\_рождения.

**Вариант 7**

1. Сортировать массив символов обменной, карманной сортировкой, сортировкой вставками . Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов массива не менее 10000.
2. Для сортированного массива строк (Фамилии) fr(1000) реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива байтовых чисел.

**Вариант 8**

1. Сортировать числовой файл(целые числа) обменной сортировкой, сортировкой вставками . Оценить реальное время на сортировку каждым из методов на текущем компьютере. Количество элементов файла не менее 10000.
2. Для сортированного массива строк (Предметы\_по специальности) pr(100) реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива целых чисел.

**Вариант 9**

1. Сортировать числовой файл(целые числа) двухпутевым слиянием . Оценить реальное время на сортировку на текущем компьютере. Количество элементов файла не менее 10000.
2. Для сортированного массива строк (Экзамены) pr(100) реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива целых чисел.

**Вариант 10**

1. Сортировать числовой массив (целые числа) методом слияний . Оценить реальное время на сортировку на текущем компьютере. Количество элементов массиване менее 10000.
2. Для сортированного массива строк (Герои\_России) hr(100) реализовать методы последовательного и бинарного поиска. Определить количество шагов необходимых для поиска введенного с клавиатуры элементов первым и вторым методом.
3. Реализовать методы последовательного и бинарного поиска для массива вещественных чисел.

**Лабораторная №3. Однонаправленные и двунаправленные списки.**

**Вариант 1**

1. Создать односвязный список для хранения Фамилии, Имени, Отчества в отдельных полях. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий городов Ханты-Мансийского автономного округа. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Создать динамический массив вещественных чисел.(С++)

**Вариант 2**

1. Создать односвязный список для хранения Фамилии, Имени, Отчества, Дата\_рождения в отдельных полях. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий городов и количества населения в них Ханты-Мансийского автономного округа. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек на основе динамического массива вещественных чисел. (С++)

**Вариант 3**

1. Создать односвязный список для хранения Фамилия, курс, группа. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий Вузов Ханты-Мансийского автономного округа. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек строк на основе односвязного линейного списка

**Вариант 4**

1. Создать односвязный список для хранения Фамилия, курс, группа, год\_поступления. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий районов Ханты-Мансийского автономного округа. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек символов на основе односвязного линейного списка

**Вариант 5**

1. Создать односвязный список для хранения Фамилия, факультет, курс. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий нефтяных местрождений Ханты-Мансийского автономного округа. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек вещественных чисел на основе односвязного линейного списка

**Вариант 6**

1. Создать односвязный список для хранения записи Фамилия, факультет, курс, группа. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения городов и количества населения в них. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек строк на основе односвязного линейного списка

**Вариант 7**

1. Создать односвязный список для хранения записи Фамилия, год рождения, факультет, курс, группа.. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий школ и количества школьников в них. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек вещественных чисел на основе односвязного линейного списка

**Вариант 8**

1. Создать односвязный список для хранения записи Фамилия, факультет, курс, группа.. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения улиц г. Нижневартовска. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек целых чисел на основе односвязного линейного списка

**Вариант 9**

1. Создать односвязный список для хранения записи Фамилия, факультет, курс, группа.. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий государств и столиц. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек байтовых чисел на основе односвязного линейного списка

**Вариант 10**

1. Создать односвязный список для хранения записи Фамилия, факультет, курс, группа, год\_поступления.. Реализовать методы: Добавить узел, удалить узел, просмотр списка, определение принадлежности элемента списку.
2. Создать двусвязный список для хранения названий наций и народностей и их количества в ХМАО. Реализовать методы: Добавить узел, удалить узел, просмотр списка слева направо и справа налево, определение принадлежности элемента списку.
3. Реализовать стек байтовых чисел на основе односвязного линейного списка

**Лабораторная №4. Сортировка. Хеширование. Динамические структуры.**

**Вариант 1**

1. Cохранить в файле Num.dat несколько целых чисел больших нуля таким образом, чтобы числа записались в сортированном виде. Для этого считать само число номером записи в файле. Удалить элементы равные 0. Использовать вспомогательный файл.
2. Создать стек для хранения Фамилий студентов. Реализовать операторы : добавить элемент, удалить элемент, вершина стека, просмотр элементов стека.
3. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодами первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя, Отчество.

**Вариант 2**

1. Cохранить в файле Num.dat несколько целых чисел больших 100 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число -100 первым номером записи в файле. Удалить элементы равные 100. Использовать вспомогательный файл.
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодами первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя.
3. Создать множество для хранения названий городов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 3**

1. Cохранить в файле Num.dat несколько целых чисел больших 1000 таким образом, чтобы числа записались в сортированном виде. Для этого считать (само число) -1000 номером записи в файле. Удалить элементы равные 1000. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодами первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения названий городов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 4**

1. Cохранить в файле Num.dat несколько целых чисел больших 500 таким образом, чтобы числа записались в сортированном виде. Для этого считать (само число) -500 номером записи в файле. Удалить элементы меньшие или равные 500. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодом первого 1 символа Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения названий городов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 5**

1. Cохранить в файле Num.dat несколько целых чисел больших 200 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число -200 номером записи в файле. Удалить элементы меньшие или равные 200. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодом первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения фамилий студентов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 6**

1. Cохранить в файле Num.dat несколько целых чисел больших 250 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число -250 номером записи в файле. Удалить элементы меньшие или равные 250. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодам и первых 3 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения фамилий студентов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 7**

1. Cохранить в файле Num.dat несколько целых чисел больших 350 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число -350 номером записи в файле. Удалить элементы меньшие или равные 350. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодам и первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения фамилий студентов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 8**

1. Cохранить в файле Num.dat несколько целых чисел больших 450 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число -450 номером записи в файле. Удалить элементы меньшие или равные 450. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодам и первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения фамилий студентов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 9**

1. Cохранить в файле Num.dat несколько целых чисел больших 550 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число номером записи в файле. Удалить элементы меньшие или равные 550. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодам и первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения фамилий студентов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Вариант 10**

1. Cохранить в файле Num.dat несколько целых чисел больших 650 таким образом, чтобы числа записались в сортированном виде. Для этого считать само число -650 номером записи в файле. Удалить элементы меньшие или равные 650. Использовать вспомогательный файл. Реализовать меню: Запись в файл. Вывод записей на экран
2. Создать хеш-функцию для хранения записей в файл в сортированном виде. Хеш- функцию связать с кодам и первых 2 символов Фамилии. Запись состоит из полей :Фамилия, Имя. Для разрешения коллизий использовать вспомогательные файлы.
3. Создать очередь для хранения фамилий студентов на основе 2 –направленного связного списка. Реализовать операторы : Добавить элемент, удалить элемент, мощность множества, просмотр всех элементов, принадлежность элемента множеству.

**Лабораторная №5. Хеширование. Разрешение коллизий заполнением пустот в файле(таблице).**

***Вариант 1***

Оцените качество хеш-функций

1. 
2. , где Fio – строка длиной 20, все символы заглавные, алфавит – кириллица.
3. , 

Постройте графики эффективности хеш-функций.

***Вариант 2***

1. Оценить качество хеш-функций

, где Fam –фамилия на русском языке, причем первая буква – заглавная.

1. Запишите данные в файлы таким образом: если не занят адрес, то данные записываются в главный файл header.txt, а при возникновении коллизий используются вспомогательные файлы 1.txt, 2.txt, 3.txt, …n.txt, где n – порядковый номер символа в алфавите.

Реализовать методы: добавить, просмотр (вывод в сортированном виде)

***Вариант 3***

1. Оценить качество хеш-функции

, где Fio – строка длиной 20, все символы заглавные, алфавит - кириллица.

1. Запишите данные в файлы таким образом: если не занят адрес, то данные записываются в главный файл student.dat, а при возникновении коллизий используются вспомогательные файлы, имя которого формируется присоединением буквы к адресу str(h):

Имя\_файла=v+str(h)+’.dat’

***Вариант 4***

Оценить качество хеш-функции



где n – количество пустых записей для каждого адреса, из интервала [0, 100]

Fio – строка Фамилия Имя Отчество, длиной 30.

Запишите данные в файл следующим образом: например, для адреса 2 в случае n=10 запись в файл начнется с адреса 1+10=11, 2+10=12

Первому адресу отводится с 0 – по 10,

Второму с 11 – по 21,

Третьему с 22 – по 32,

Четвертому с 33 – по 43 и т.д.

Pord(Fio[1]) – это порядковый номер символа в алфавите.

‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| а | | б | в | | г | | д | | е | | ё | | ж | | З | | и | | | й | К | л | | м | | н | | о | | п |
| 1 | | 2 | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | | 11 | 12 | 13 | | 14 | | 15 | | 16 | | 17 |
| р | с | | | т | | у | | ф | | х | | ц | | ч | | Ш | | щ | ъ | | Ы | | ь | | э | | ю | | я | |
| 18 | 19 | | | 20 | | 21 | | 22 | | 23 | | 24 | | 25 | | 26 | | 27 | 28 | | 29 | | 30 | | 31 | | 32 | | 33 | |

Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов (в сортированном виде).

Ответьте на вопросы

1. Эффективна ли данная функция для хранения 330, 1000, 10000 записей.
2. Как изменить функцию, чтобы сохранить 200 фамилий, начинающихся с буквы «А», и 100 фамилий с буквы «Я», остальные 50 меньше 100 раз

***Вариант 5***

1. Оцените качество хеш-функции

, где Pord(s) - порядковый номер символа в алфавите ‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.

1. Запишите данные в файл mydb.txt. Для разрешения коллизий используйте заполнение пустот.
2. Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов.

***Вариант 6***

1. Оцените качество хеш-функции

, где Pord(s) - порядковый номер символа в алфавите ‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.

1. Запишите данные в файл person.txt. Для разрешения коллизий используйте заполнение пустот в файле.
2. Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов.

***Вариант 7***

1. Оцените качество хеш-функции

, где (s) - порядковый номер символа в алфавите ‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.

1. Запишите данные в файл person.txt. Для разрешения коллизий используйте заполнение пустот в файле.
2. Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов.

***Вариант 8***

1. Оцените качество хеш-функции

, где  - порядковый номер символа в алфавите ‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.

1. Запишите данные в файл person.txt. Для разрешения коллизий используйте заполнение пустот в файле.
2. Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов.

***Вариант 9***

1. Оцените качество хеш-функции

, где  - порядковый номер символа в алфавите ‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.

1. Запишите данные в файл person.txt. Для разрешения коллизий используйте заполнение пустот в файле.
2. Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов.

***Вариант 10***

1. Оцените качество хеш-функции

, где  - порядковый номер символа в алфавите ‘A’,’a’ – 1; ‘Б’,’б’ – 2; ‘В’,’в’ – 2; ‘Г’,’г’ – 4 и т.д.

1. Запишите данные в файл person.txt. Для разрешения коллизий используйте заполнение пустот в файле.
2. Реализуйте операторы: добавить элемент, поиск элемента, просмотр элементов.

**Лабораторная №6.**

**Хеширование. Разрешение коллизий с использованием** **вспомогательных файлов**

***Вариант 1***

1. Для хранения записей в файле используете хеш-функцию.  Запись имеет структуру Fio:string[20], dr: string[12]
2. Для разрешения коллизий используйте вспомогательные файлы, имена которых связаны с адресом, а именно имя файла = str(адрес)+’.dat’. Для избежания ошибки открытия вспомогательных файлов заведите log-файл, в котором построчно сохраняются имена файлов. Если получившееся имя вспомогательного файла имеется в log-файле, то применяется процедура Reset(t), в противном случае – Rewrite(t).
3. Реализуйте операторы: добавить запись, поиск записи, просмотр всех значений.

***Вариант 2***

1. Для хранения записей в файле используете хеш-функцию  Запись имеет структуру Fio:string[20]; kurs, group: byte
2. Для разрешения коллизий используйте вспомогательные файлы, имена которых связаны с адресом, а именно имя файла = str(адрес)+’.dat’. Для избежания ошибки открытия вспомогательных файлов заведите log-файл, в котором построчно сохраняются имена файлов. Если получившееся имя вспомогательного файла имеется в log-файле, то применяется процедура Reset(t), в противном случае – Rewrite(t).
3. Реализуйте операторы: добавить запись, поиск записи, просмотр всех значений.

***Вариант 3***

1. Для хранения записей структуры: Fam: string[20];

Name: string[10];

Otch: string[15];

Dr: string[12];

в файле используйте хеш-функцию: 

1. Для разрешения коллизий используйте вспомогательные файлы.
2. Реализуйте операторы: добавить запись, поиск записи, просмотр всех значений.

***Вариант 4***

1. Для хранения записей структуры: City\_name: string[20];

Region\_name: string[30];

City\_peoples: longint;

в файле используйте хеш-функцию: 

1. Для разрешения коллизий используйте вспомогательные файлы, имена которых связаны с адресом, а именно имя файла = str(адрес)+’.dat’. Для избежания ошибки открытия вспомогательных файлов заведите log-файл, в котором построчно сохраняются имена файлов. Если получившееся имя вспомогательного файла имеется в log-файле, то применяется процедура Reset(t), в противном случае – Rewrite(t).
2. Реализуйте операторы: добавить запись, поиск записи, вывод городов, начинающихся с определенной буквы, просмотр всех значений.

***Вариант 5***

1. Для хранения записей структуры: Название\_предприятия: string[20];

Количество\_работников: longint;

в файле используйте хеш-функцию: , где S – название предприятия.

1. Для разрешения коллизий используйте вспомогательные файлы.
2. Реализуйте операторы: добавить запись, поиск записи, просмотр всех значений

***Вариант 6***

1. Для хранения записей структуры: student= record

Fam: string[20];

Dr: string[20];

Hob1:string[12];

Hob2:string[12];

Hob3:string[12];

Hob4:string[12];

в файле используйте хеш-функцию: 

1. Определить, что необходимо выполнять, чтобы адресация началась с нуля.
2. Реализуйте операторы: добавить запись, поиск записи по полю Fam, просмотр всех элементов.

***Вариант 7***

1. Для хранения записей структуры: student= record

Fam: string[20];

Dr: string[20];

Hob1:string[12];

Hob2:string[12];

Hob3:string[12];

Hob4:string[12];

в файле используйте хеш-функцию: 

1. Определить, что необходимо выполнять, чтобы адресация началась с нуля.
2. Реализуйте операторы: добавить запись, поиск записи по полю Fam, просмотр всех элементов.

***Вариант 8***

1. Для хранения записей структуры: student= record

Fam: string[20];

Dr: string[20];

Hob1:string[12];

Hob2:string[12];

Hob3:string[12];

Hob4:string[12];

в файле используйте хеш-функцию: 

1. Определить, что необходимо выполнять, чтобы адресация началась с нуля.
2. Реализуйте операторы: добавить запись, поиск записи по полю Fam, просмотр всех элементов.

***Вариант 9***

1. Для хранения записей структуры: student=record

Fam: string[20];

Dr: string[20];

Hob1:string[12];

Hob2:string[12];

Hob3:string[12];

Hob4:string[12];

в файле используйте хеш-функцию: 

1. Определить, что необходимо выполнять, чтобы адресация началась с нуля.
2. Реализуйте операторы: добавить запись, поиск записи по полю Fam, просмотр всех элементов.

**Лабораторная №7** **Деревья. Сортировка на двоичном дереве.**

Используя двоичное дерево сортировать массив

|  |  |
| --- | --- |
| ***Вариант*** | ***Массив*** |
| ***1*** | 5, 9, 13, 14, 2, 7, 1, 15, 18, 8, 4, 3, 50 |
| ***2*** | 4, 5, 13, 8, 41, 52, 41, 64, 52, 17, 8, 9, 30 |
| ***3*** | 5, 1, 91, 4, 7, 51, 40, 61, 57, 71, 9, 8, 3 |
| ***4*** | 15, 3, 46, 54, 17, 81, 56, 72, 61, 59, 8, 3, 21 |
| ***5*** | 54, 31, 72, 62, 5, 95, 7, 64, 59, 87, 63, 64 |
| ***6*** | 57, 91, 64, 52, 73, 61, 5, 91, 11, 13, 15, 105, 7 |
| ***7*** | 17, 54, 64, 2, 17, 52, 1, 19, 17, 64, 52, 17, 24 |
| ***8*** | 54, 101, 27, 14, 17, 18, 54, 91, 72, 76, 75, 102 |
| ***9*** | 31, 102, 127, 114, 19, 54, 17, 88, 77, 11, 4, 7 |
| ***10*** | 5, 102, 7, 64, 8, 92, 54, 17, 25, 29, 44, 71 |

1. Реализуйте дерево в виде двусвязного списка
2. Обход слева, обход справа, симметрический обход.
3. Реализуйте программу сортировки массива на дереве методом симметрического обхода.