

4 лаба

Задание: Реализовать программу для простейшего моделирования **упорядоченного линейного списка с помощью массива**. Должны быть реализованы все основные действия:

- проход по списку с выводом на экран информационных частей элементов
- поиск элемента с заданной информационной частью
- добавление нового элемента (т.к. список **упорядоченный**, **то** программа должна находить место для нового элемента, т.е. поиск должен останавливаться при обнаружении первого элемента, большего заданного)
- удаление заданного элемента со сдвигом (при необходимости) хвостовой части влево для заполнения образовавшейся пустой ячейки.

5 лаба

Задание: Реализовать **линейный список на базе массива с указателями-индексами**. Список должен иметь заголовки (нулевая ячейка массива) с номером первого элемента списка. Набор операций - стандартный.

6 лаба

Задание: Реализовать **линейный динамический список** со стандартным набором операций:

- проход по списку с выводом на экран информационных частей элементов
- поиск элемента с заданной информационной частью
- проверка пустоты списка
- добавление элемента
- удаление элемента: удаляемый из основного списка элемент добавляется во вспомогательный список с возможностью просмотра вспомогательного списка.

4-5-6 Лабы. Теория

- Линейный список – это набор связанных **однотипных** элементов, в котором каждый элемент каким-то образом определяет следующий за ним элемент. В отличие от стека и очереди, добавление нового элемента возможно в **любом месте** списка, также можно удалить **любой** элемент списка.

7 лаба

Задание: Реализовать **линейный динамический двунаправленный список** со следующим набором операций:

- просмотр списка в прямом и обратном направлениях
- поиск заданного элемента в прямом и обратном направлениях
- добавление элемента перед или после заданного
- удаление заданного элемента

Список должен иметь заголовок и быть кольцевым.

8 лаба

Задание: Реализовать набор подпрограмм для выполнения основных операций со **списком списков**.

Основные операции:

- полный проход по всей структуре
- поиск заданного элемента
- добавление нового элемента в массив с пустым связанным списком
- добавление нового элемента в связанный список
- удаление элемента из связанного списка
- удаление элемента из базового массива

7-8 Лабы. Теория

Двунаправленные списки: каждый элемент "знает" **обоих** своих соседей, как левого, так и правого.

Для этого каждый элемент должен иметь не одно, а два связующих поля: указатель на элемент слева и указатель на элемент справа.

9 лаба

Задание. Построение и обход идеально сбалансированных двоичных деревьев. Реализовать программу, выполняющую следующий набор операций:

- построение идеально сбалансированного двоичного дерева с заданным числом вершин
- построчный вывод дерева на основе процедуры обхода в прямом порядке
- построчный вывод дерева на основе процедуры обхода в симметричном порядке
- построчный вывод дерева на основе процедуры обхода в обратно-симметричном порядке

10 лаба

Задание: Добавить в предыдущую программу **нерекурсивный** вариант процедуры обхода дерева в симметричном порядке.

11 лаба

Задание: Обработка произвольных двоичных деревьев. Реализовать программу, выполняющую следующий набор операций с двоичными деревьями:

- поиск вершины с заданным значением информационной части
- добавление левого или правого потомка для заданной вершины
- построчный вывод дерева с помощью основных правил обхода
- уничтожение всего дерева

12 лаба

Задание: Построение и обработка двоичных деревьев поиска. Реализовать программу, выполняющую следующий набор операций с деревьями поиска:

- поиск вершины с заданным значением ключа с выводом счетчика числа появлений данного ключа
- добавление новой вершины в соответствии со значением ее ключа или увеличение счетчика числа появлений
- построчный вывод дерева в наглядном виде с помощью обратно- симметричного обхода

- вывод всех вершин в одну строку по порядку следования ключей с указанием для каждой вершины значения ее счетчика появлений
- удаление вершины с заданным значением ключа

9-12 Лабы. Теория

ИДЕАЛЬНО СБАЛАНСИРОВАННОЕ ДЕРЕВО (ИСД)

Двоичное Дерево называется идеально сбалансированным, если для **каждой** вершины число вершин в левом и правом ее поддеревьях отличаются не более чем на единицу. И данное условие должно выполняться для всех вершин дерева!

ПРАВИЛА ОБХОДА:

- **прямой обход:** корневая вершина, левый потомок, правый потомок
- **симметричный:** левый потомок, корневая вершина, правый потомок
- **обратный обход:** левый потомок, правый потомок, корневая вершина

ПРОИЗВОЛЬНОЕ ДВОИЧНОЕ ДЕРЕВО - любое двоичное дерево, не должно удовлетворять условию идеально сбалансированного

ДЕРЕВО ПОИСКА

Двоичное дерево называется деревом поиска или поисковым деревом, если для **каждой** вершины дерева все ключи в ее **левом** поддереве **меньше** ключа этой вершины, а все ключи в ее **правом** поддереве **больше** ключа вершины.