Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №11.**

**«Динамические массивы»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

1. Написать функцию для создания списка. Функция может создавать пустой список, а затем добавлять в него элементы.
2. Написать функцию для печати списка. Функция должна предусматривать вывод сообщения, если список пустой.
3. Написать функции для удаления и добавления элементов списка в соответствии со своим вариантом.
4. Выполнить изменения в списке и печать списка после каждого изменения.
5. Написать функцию для записи списка в файл.
6. Написать функцию для уничтожения списка.
7. Записать список в файл, уничтожить его и выполнить печать (при печати должно быть выдано сообщение "Список пустой").
8. Написать функцию для восстановления списка из файла.
9. Восстановить список и распечатать его.
10. Уничтожить список.

Записи в линейном списке содержат ключевое поле типа \*char(строка символов). Сформировать двунаправленный список. Удалить элемент с заданным ключом. Добавить К элементов перед элементом с заданным номером.

Реализовать однонаправленный и двунаправленный список, стек, очередь.

**Анализ задачи**

**Основные операции**

Для всех линейных структур данных (списки, стеки, очереди) характерны следующие операции:

* **Инициализация**: Создание пустой структуры данных.
* **Добавление элемента**:
  + В начало (для стека и списка).
  + В конец (для очереди и списка).
* **Удаление элемента**:
  + По ключу (для списка).
  + Из начала (для стека и очереди).
* **Вставка элемента**: Вставка элемента на определенную позицию (для списка).
* **Печать элементов**: Вывод всех элементов структуры данных.
* **Сохранение в файл**: Запись элементов структуры данных в файл для последующего восстановления.
* **Восстановление из файла**: Чтение элементов из файла и восстановление структуры данных.
* **Очистка структуры**: Удаление всех элементов из структуры данных.

Многие функции, такие как сохранение в файл, восстановление из файла и печать элементов, являются общими для всех линейных структур данных. Например:

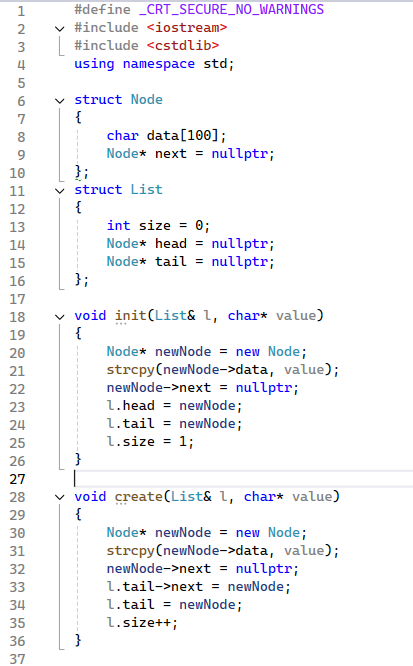
* **Сохранение в файл**: Для всех структур данных можно использовать функцию, которая последовательно записывает элементы в файл.
* **Восстановление из файла**: Функция, которая читает данные из файла и восстанавливает структуру данных.
* **Печать элементов**: Функция, которая проходит по всем элементам структуры и выводит их на экран.

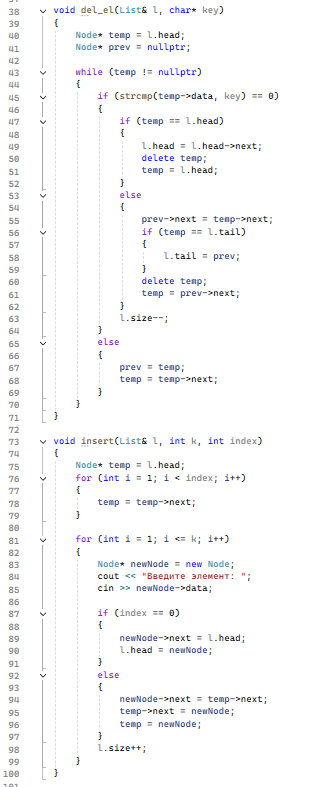
#### ****Особенности реализации****

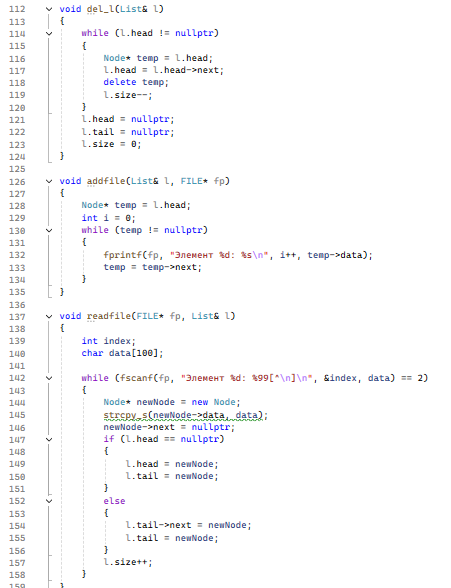
* **Списки**: Поддерживают операции вставки и удаления элементов в любом месте списка, а также доступ к элементам по индексу.
* **Стеки**: Работают по принципу LIFO (Last In, First Out), поддерживают добавление и удаление элементов только с одного конца (вершины стека).
* **Очереди**: Работают по принципу FIFO (First In, First Out), поддерживают добавление элементов в конец и удаление из начала.

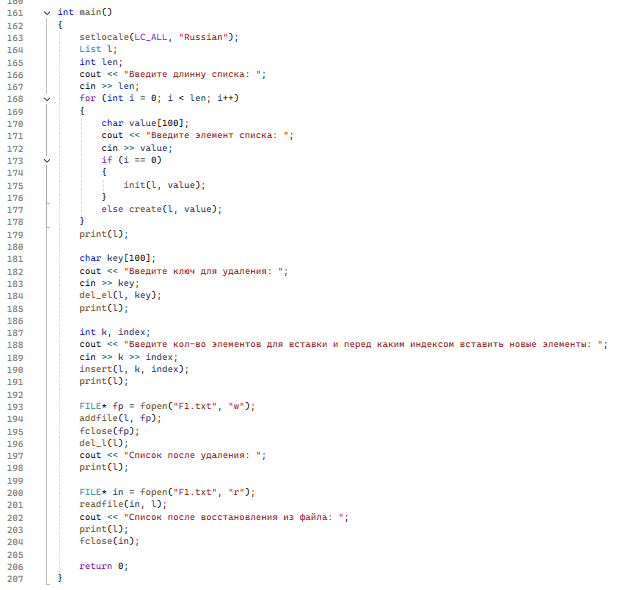
Код программы.

Односвязный список.

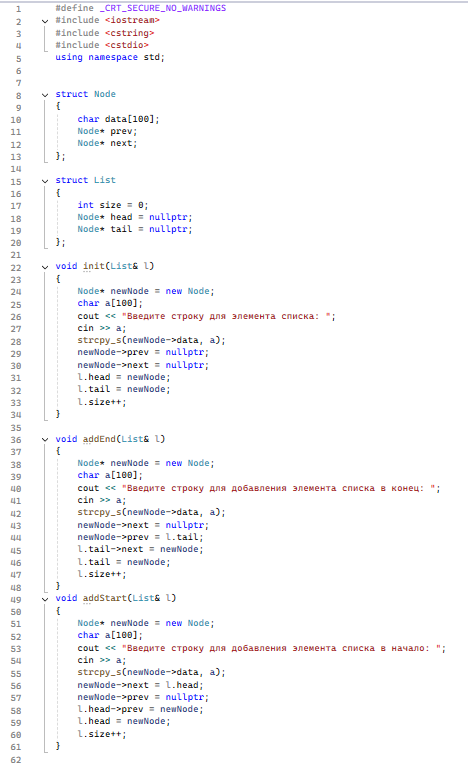


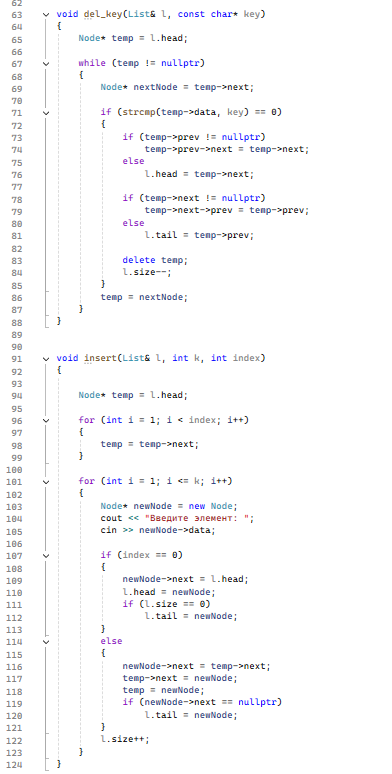
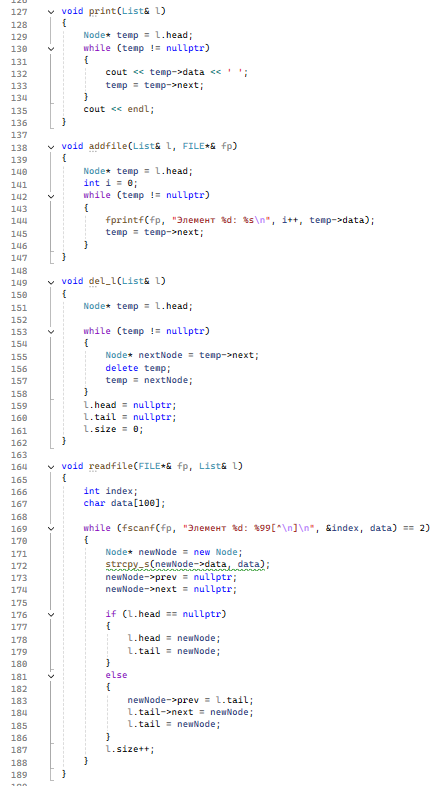


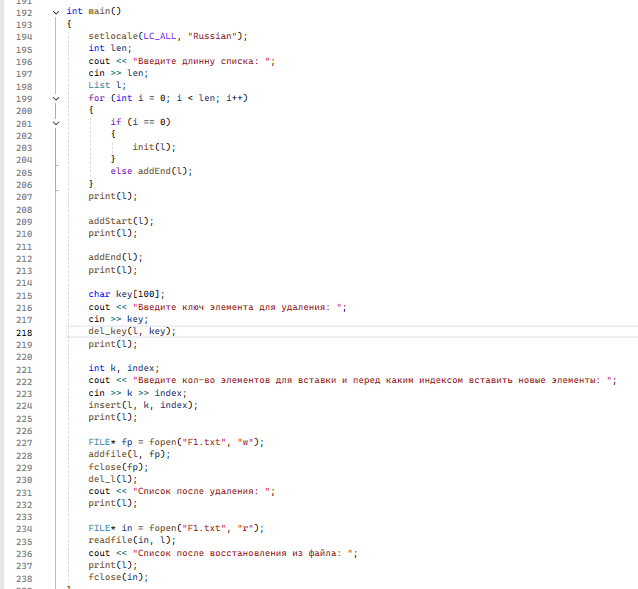


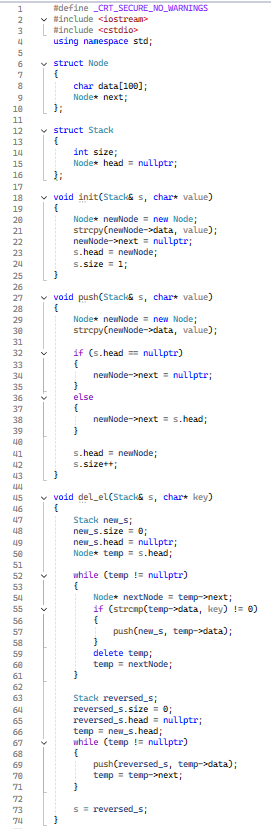


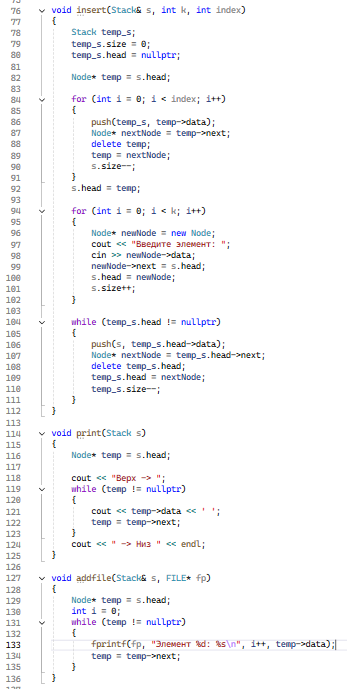
Двусвязный список

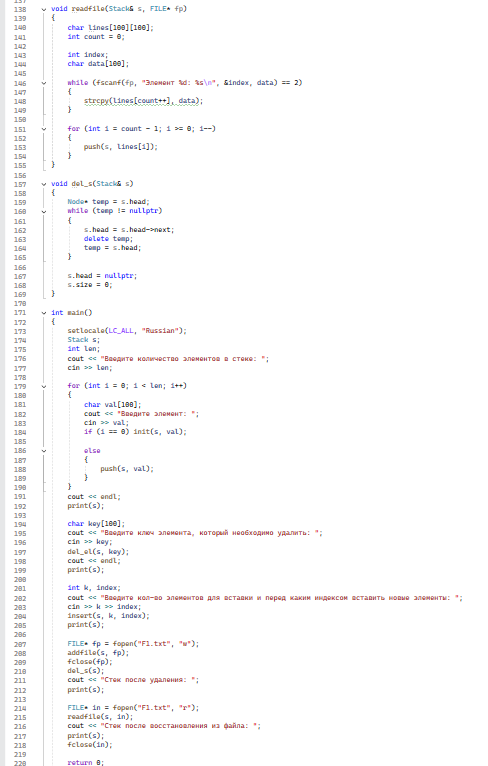




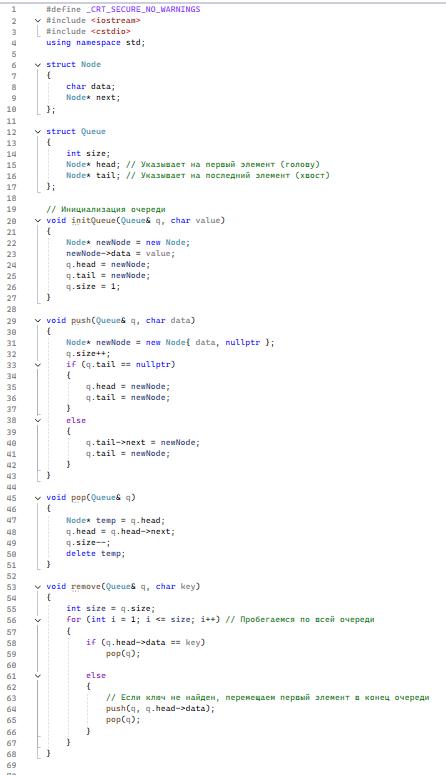


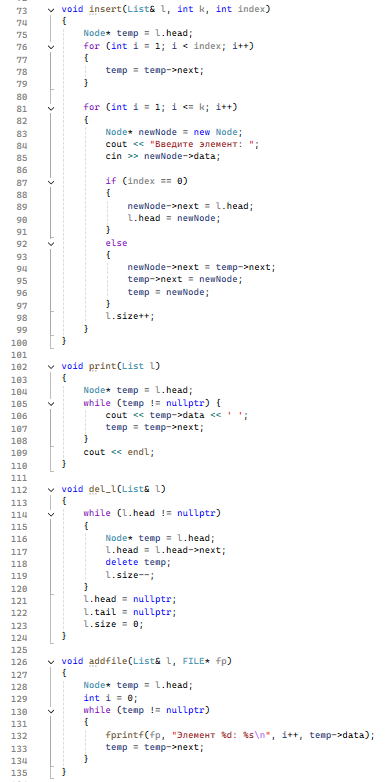
Стек.

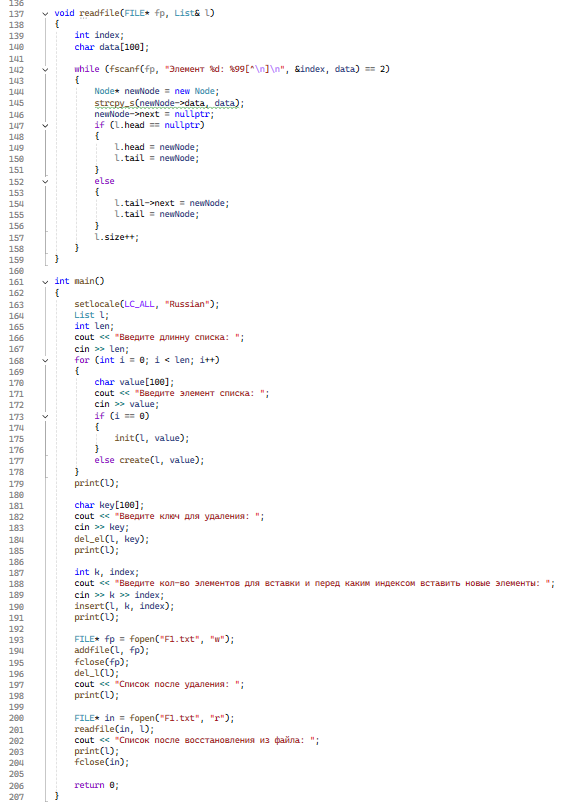




Очередь.

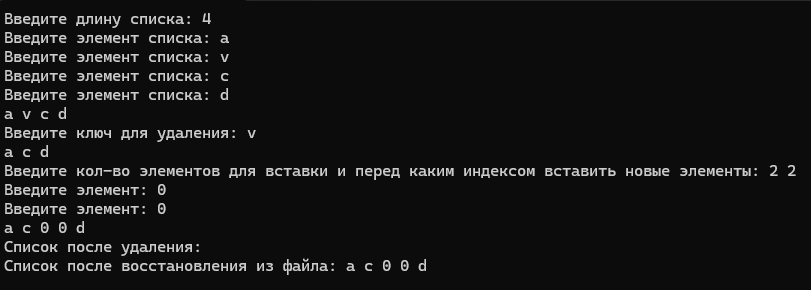




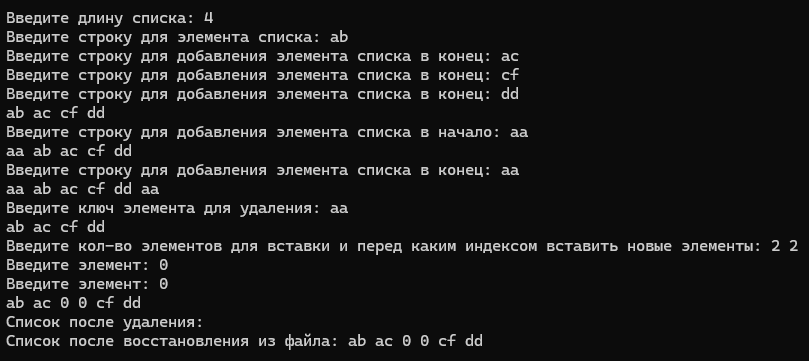


Вывод программы.

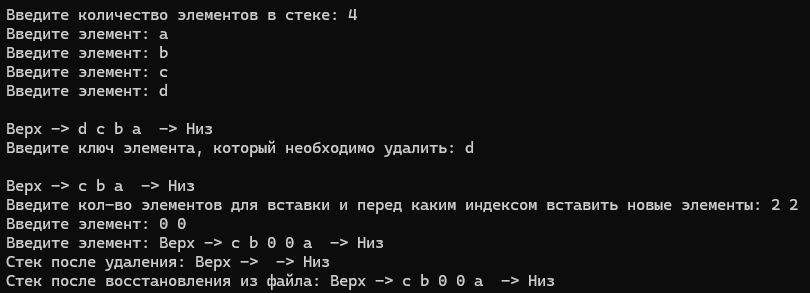
Односвязный список.



Двусвязный список.



Стек.



Очередь.

