Пермский национальный исследовательский политехнический университет ПНИПУ

Отчёт по лабораторной работе на тему “Cписки, стеки, очереди”

Выполнил студент группы РИС-23-3Б:

Смирнов Андрей Сергеевич

Проверила:

Доцент кафедры ИТАС

О.А. Полякова

2024

1. Постановка задачи:
2. Создание списка.
3. Добавление элемента в список
4. Удаление элемента из списка (в соответствии со своим вариантом).
5. Печать списка.
6. Запись списка в файл.
7. Уничтожение списка.
8. Восстановление списка из файла.
9. Анализ задачи:

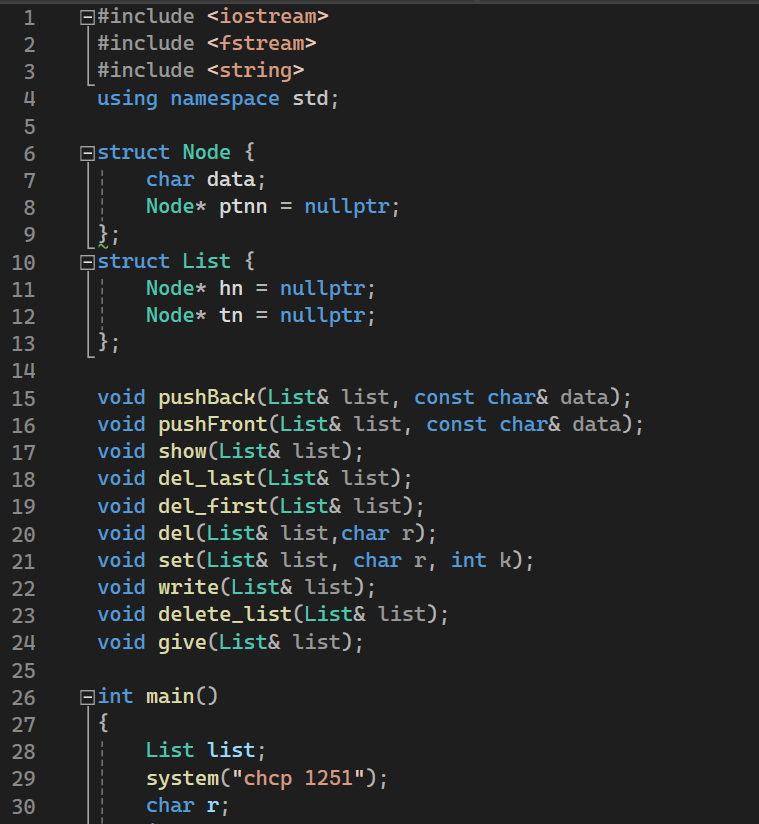
Необходимо создать список, затем заполнить его n количеством знаков, вводимых пользователем. Ввод будет обрабатываться через функцию.

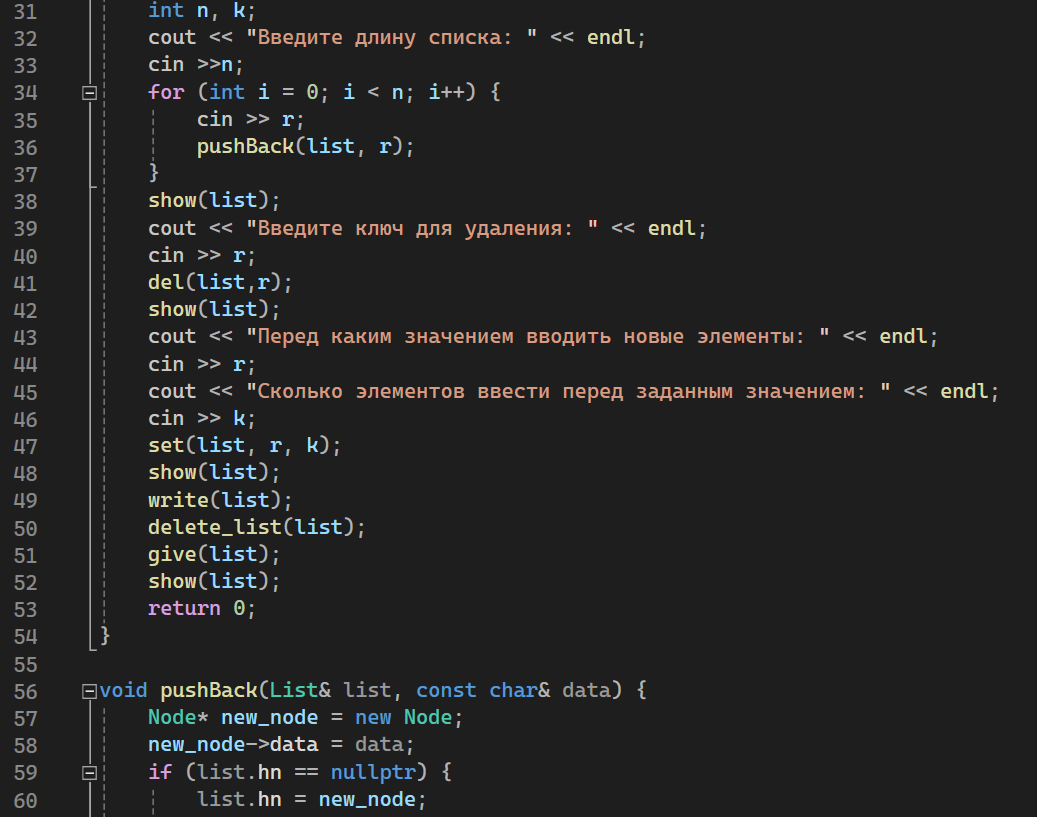
Для удаления блока по значению пройдём по значениям от головы списка до последнего блока, если блок с заданным ключом не будет найден, вывести что такого в списке нет. Удаление происходит так: голова доходит до нужного блока, указатель с головы перекидывается на следующий, блок удаляется. Запись файла происходит, как и вывод через функцию, с первого блока списка в файл записываются значения.

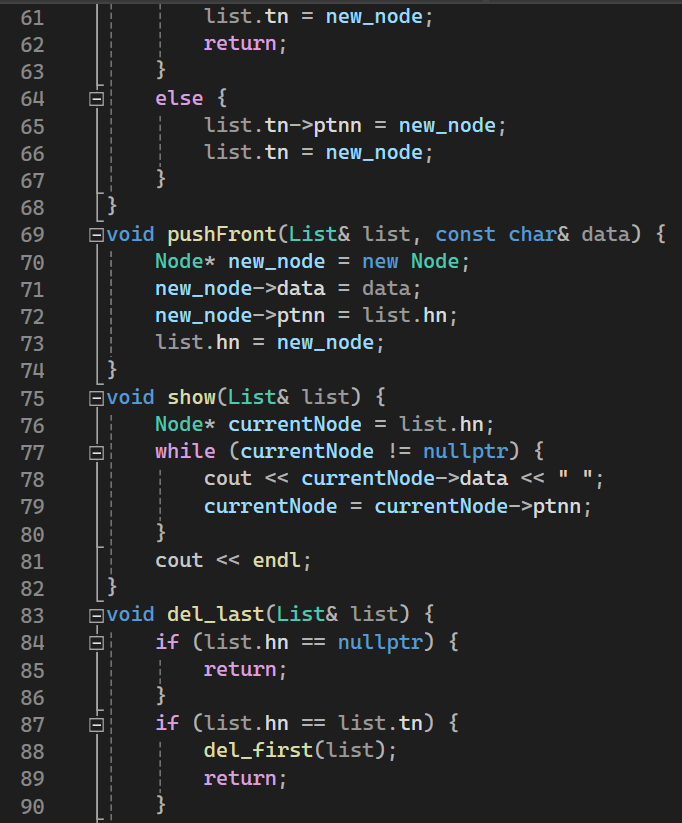
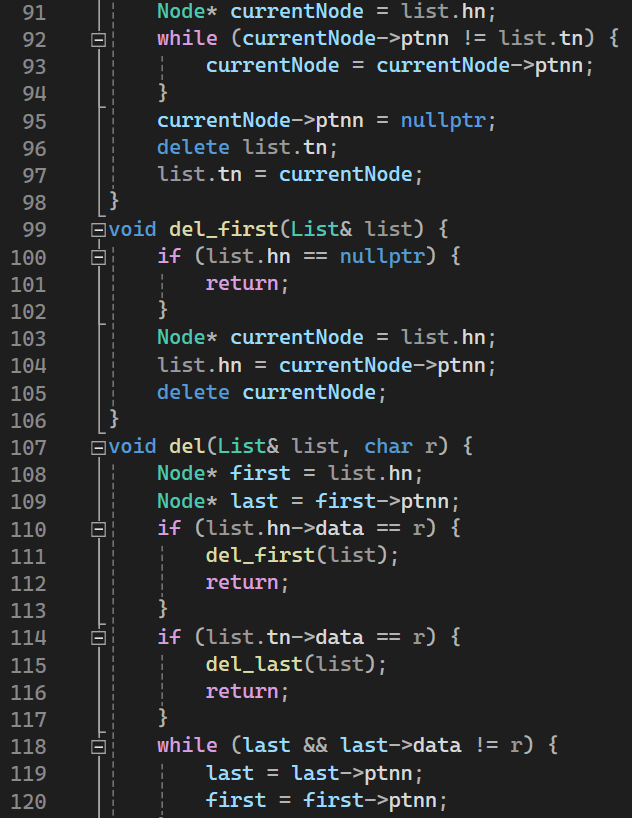
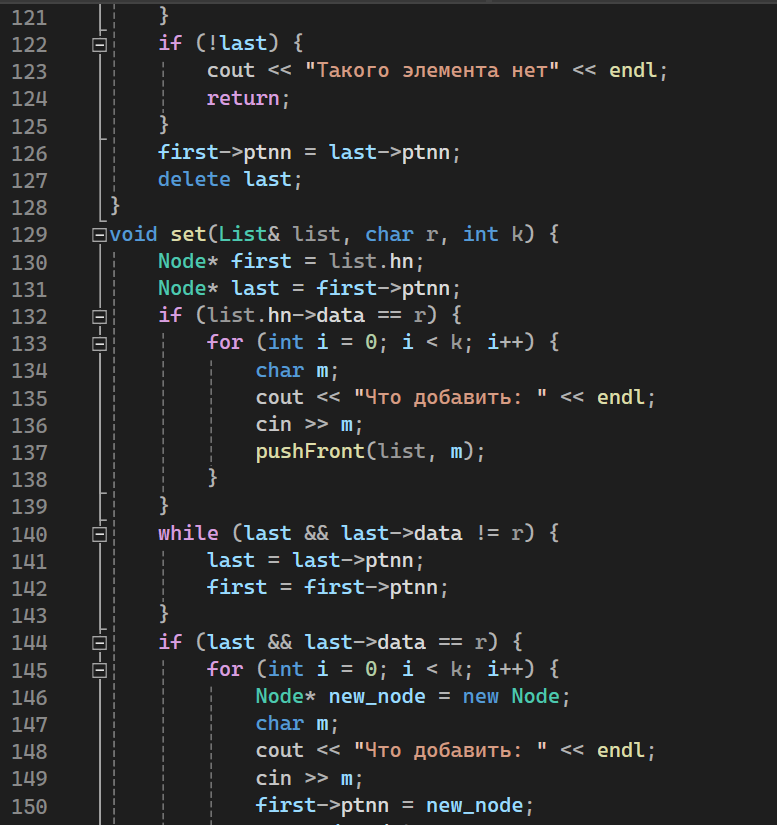
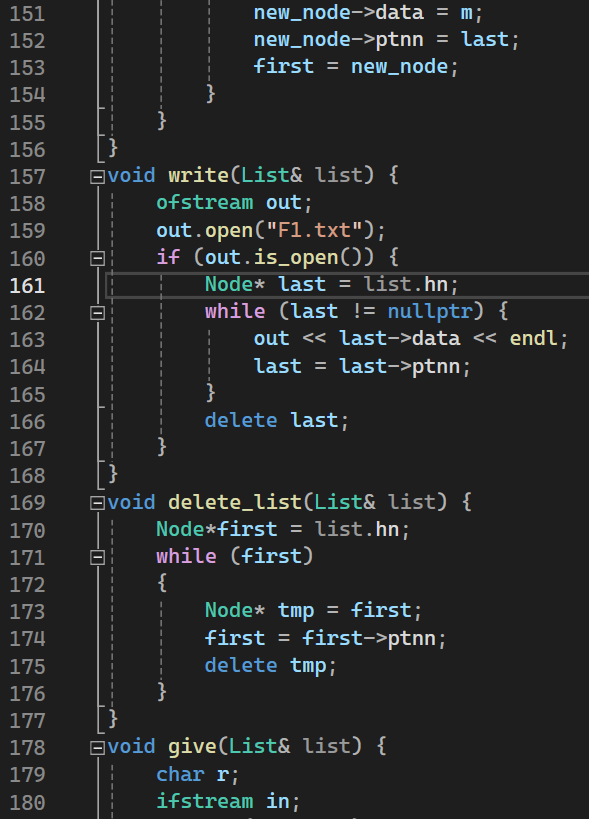
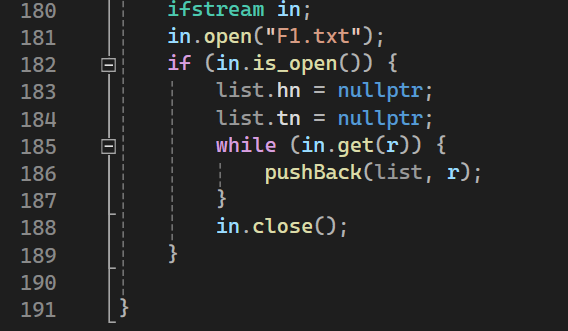
Удаление списка похоже на вывод: с первого блока удаляется блок, а следующий блок приравнивается к временной переменной, указывающей на следующий блок.

Восстановление списка происходит построчно: из файла построчно берётся значение и присваивается блоку списка.

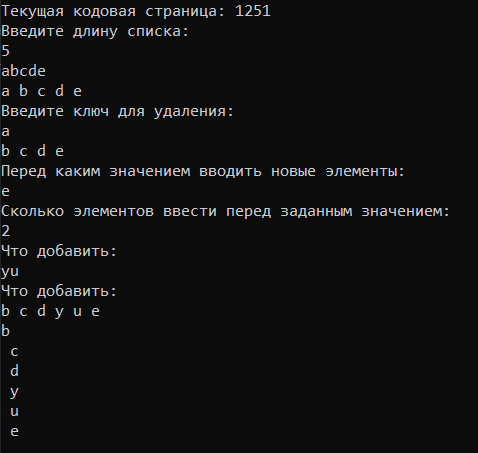
1. Блок-схема:
2. Код:

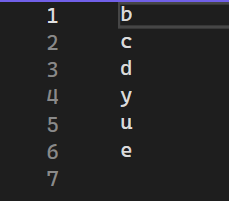




Результат работы:





6)Вывод: Списки удобны для изменения данных в середине списка, они быстрее обрабатываются по сравнению с массивами, но сложнее в создании.

7) GitHub:

<https://github.com/Andr0medA007/Labs>

Двумерный массив:

1. Постановка задачи:
2. Создание списка.
3. Добавление элемента в список
4. Удаление элемента из списка (в соответствии со своим вариантом).
5. Печать списка.
6. Запись списка в файл.
7. Уничтожение списка.
8. Восстановление списка из файла.
9. Анализ задачи:

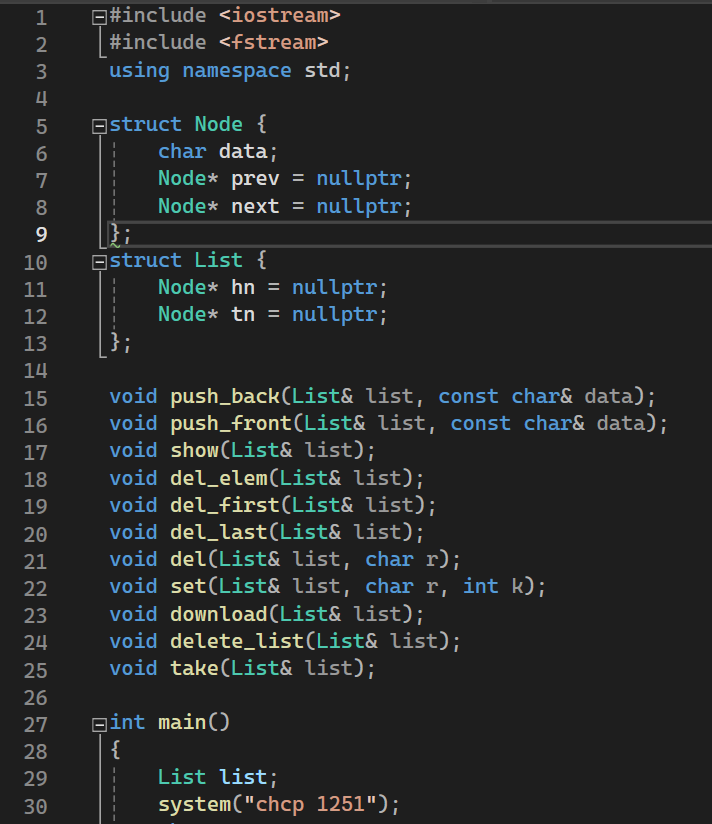
Необходимо создать список, затем заполнить его n количеством знаков, вводимых пользователем. Ввод будет обрабатываться через функцию.

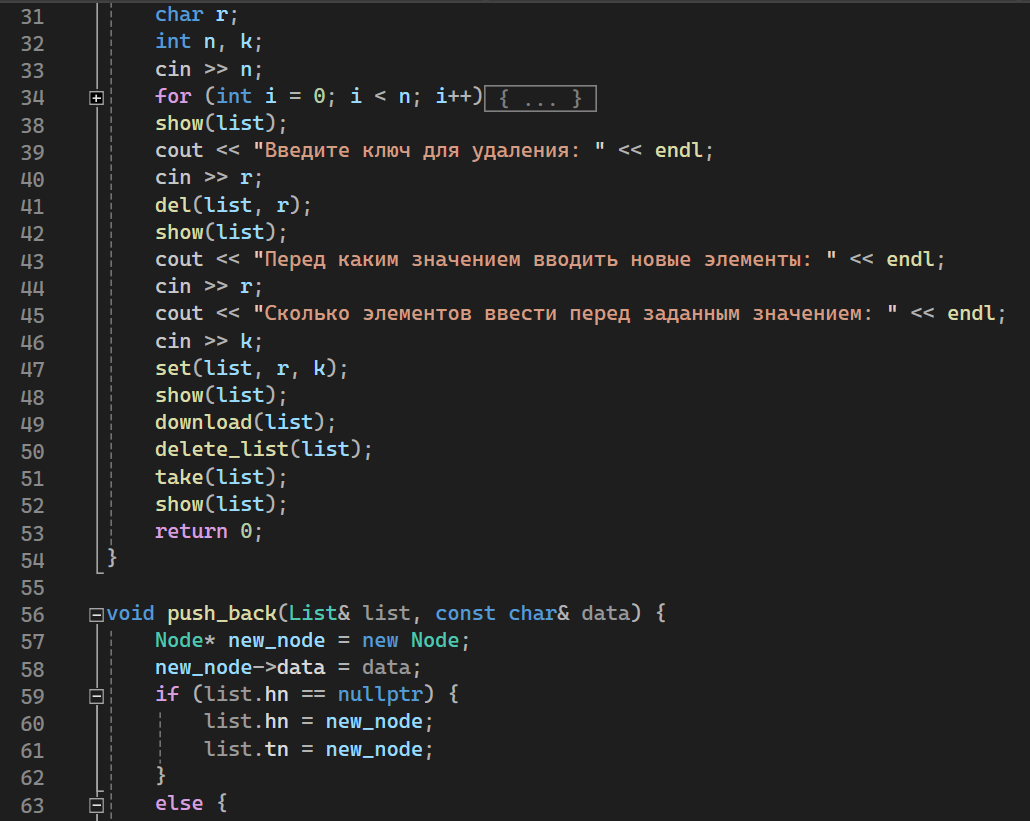
Для удаления блока по значению пройдём по значениям от головы списка до последнего блока, если блок с заданным ключом не будет найден, вывести что такого в списке нет. Удаление происходит так: голова доходит до нужного блока, указатель с головы перекидывается на следующий, блок удаляется. Запись файла происходит, как и вывод через функцию, с первого блока списка в файл записываются значения.

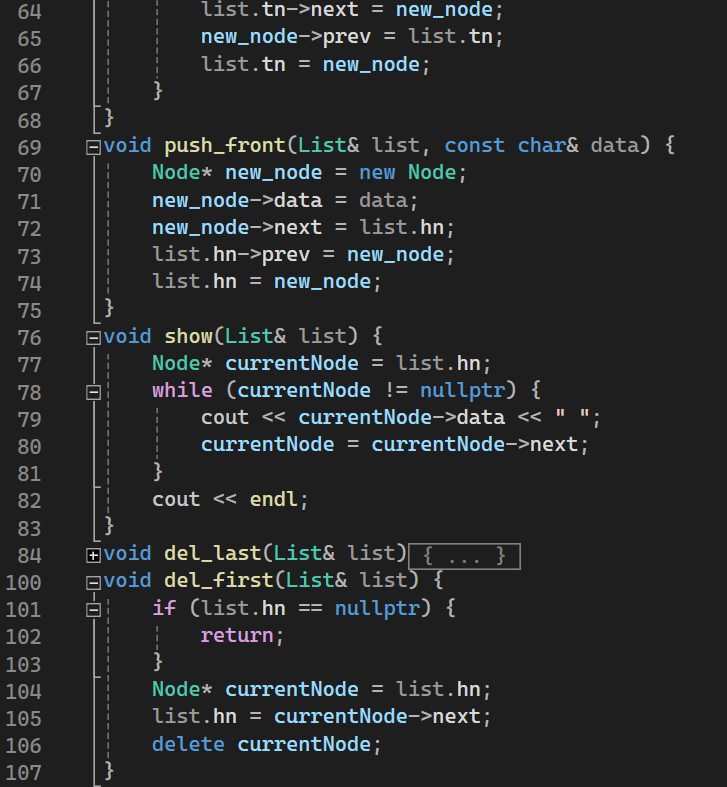
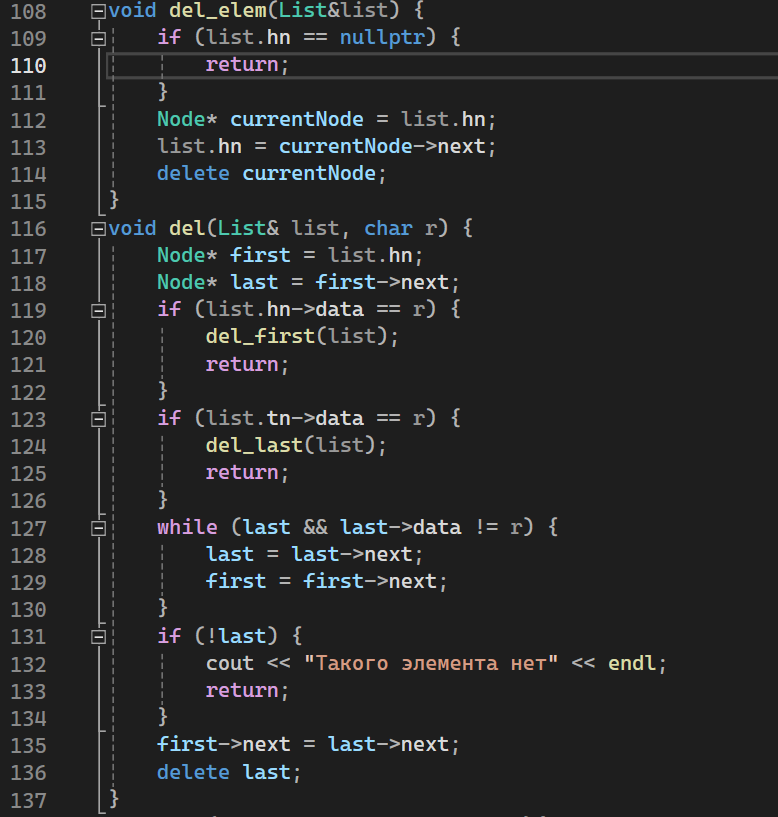
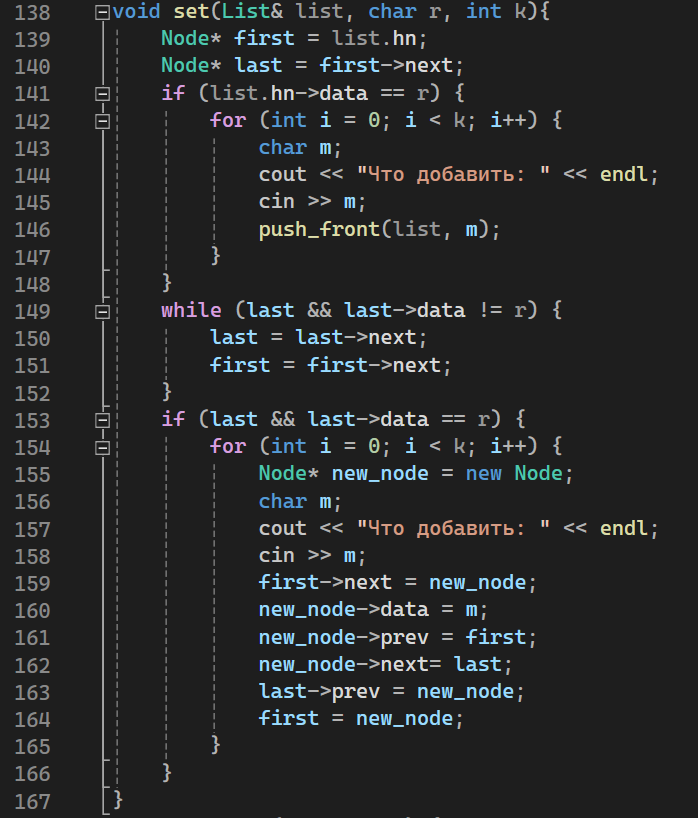
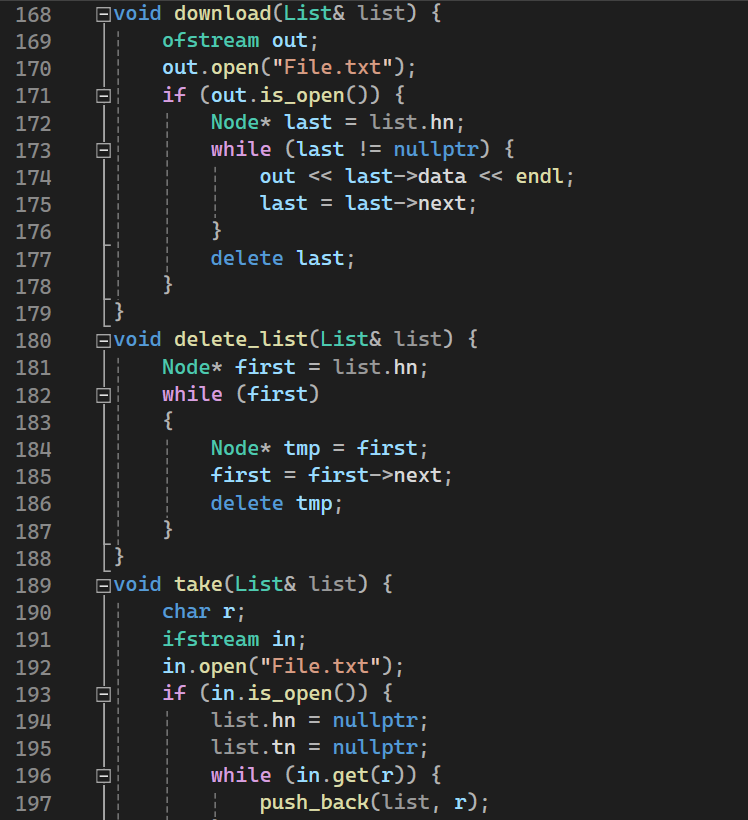
Удаление списка похоже на вывод: с первого блока удаляется блок, а следующий блок приравнивается к временной переменной, указывающей на следующий блок.

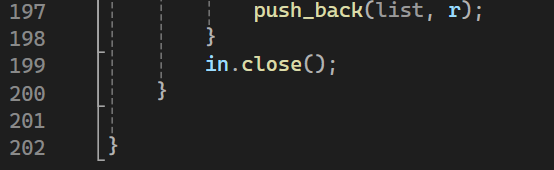
Восстановление списка происходит построчно: из файла построчно берётся значение и присваивается блоку списка.

1. Блок-схема:
2. Код:

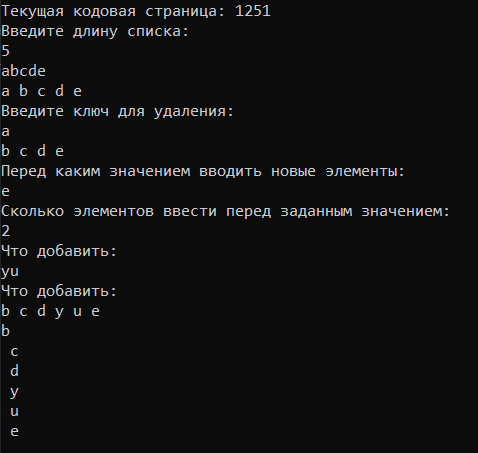


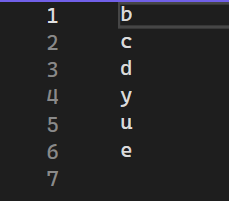




Результат работы:





6)Вывод: Списки удобны для изменения данных в середине списка, они быстрее обрабатываются по сравнению с массивами, но сложнее в создании.

7) GitHub:

<https://github.com/Andr0medA007/Labs>