Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Высшая школа кибертехнологий, математики и статистики

Кафедра информатики

Направление 38.03.05 Бизнес-информатика

Профиль «Цифровая трансформация бизнеса»

**ОТЧЕТ**

**По выполнению практической работы №2**

Выполнила:

студент группы 15.27Д-БИ19/22б

3 курса ВШКМиС

Нгуен Као Бач

Москва-2024

1. **Формулировка задачи варианта 14:**

https://onlinegdb.com/QwikQ\_HTL

Дан линейный однонаправленный список L

1. Разработать функцию, которая создает из значений узлов списка L два новых списка: L1 – из положительных элементов массива L; L2 – из остальных элементов списка L.

2. Разработать функцию, которая удаляет из списка L2 все отрицательные элементы.

3. Разработать функцию, которая в списке L1 узел с максимальным значением размещает перед первым узлом.

**2.Определение списка операций над списком, которые выявлены в процессе исследования задач дополнительного задания.**

2.1 Определить структуру узла однонаправленного списка в соответствии с вариантом.

struct Node {

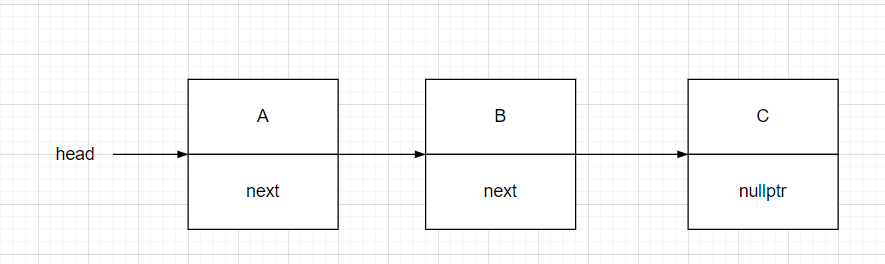
int data;

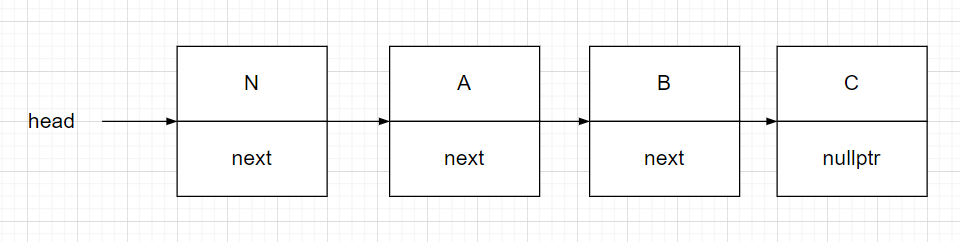
Node\* next;

};

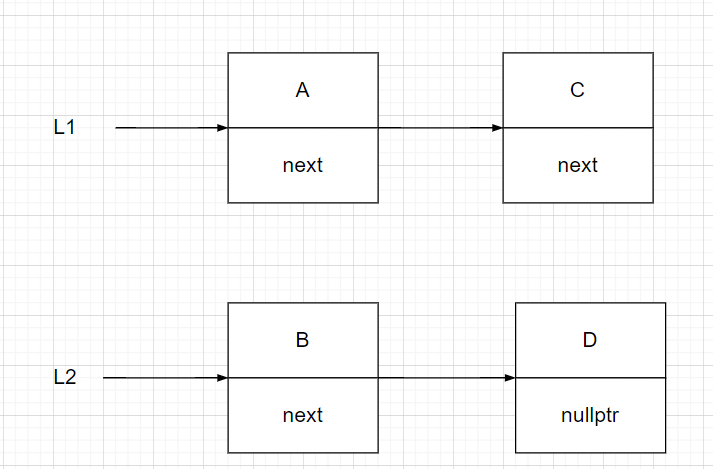
2.2 Изобразить (рисунок) для каждой операции полученного списка процесс выполнения операции на существующем однонаправленном списке.

А. Вставьте кнопку в начало списка

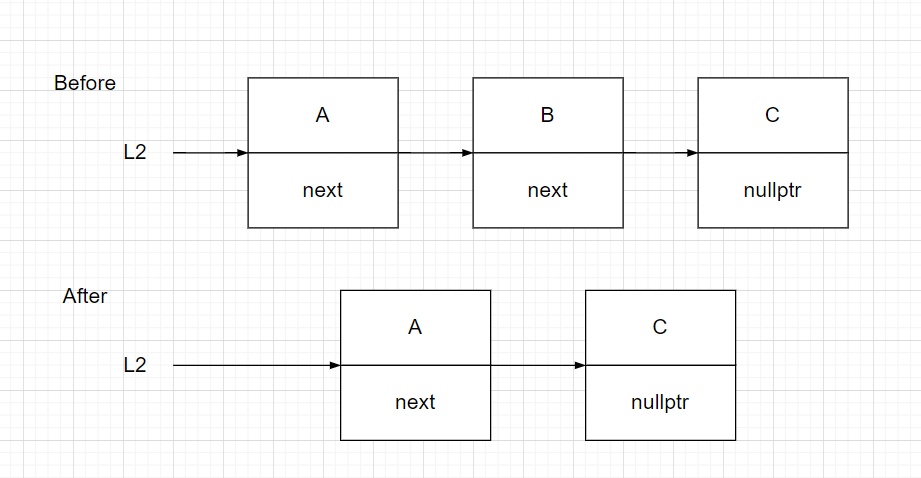
Перед вставкой

После вставки нового узла N в исходный список

**B. Разделение списка на L1 и L2**

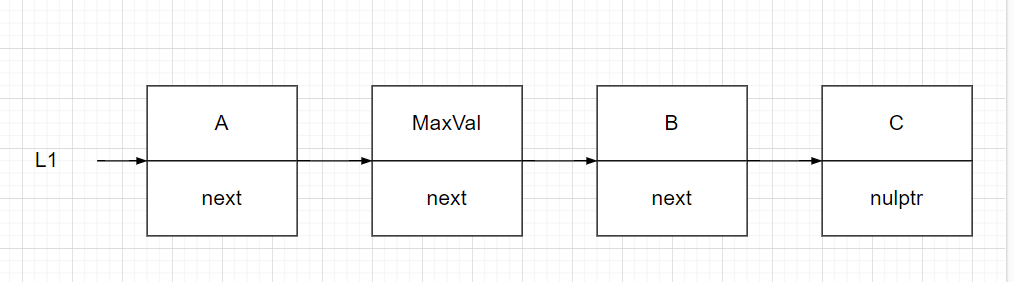
Разделение исходный список на L1 (содержащий положительные числа) и L2 (оставшиеся элементы). Предположим, что A и C положительны.

**C. Удаление отрицательных элементов из L2: Если B < 0**

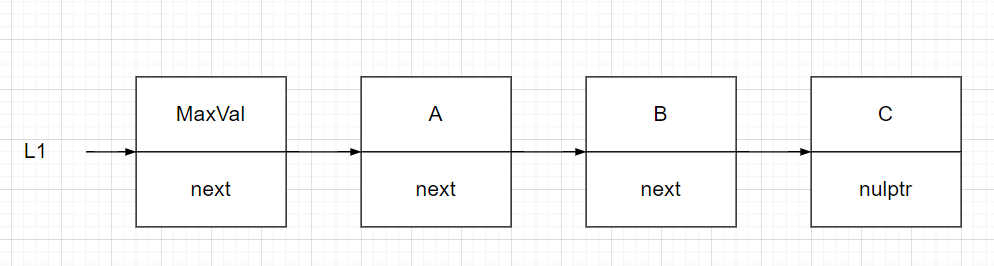


**D. Перемещение максимального элемента в начало L1**

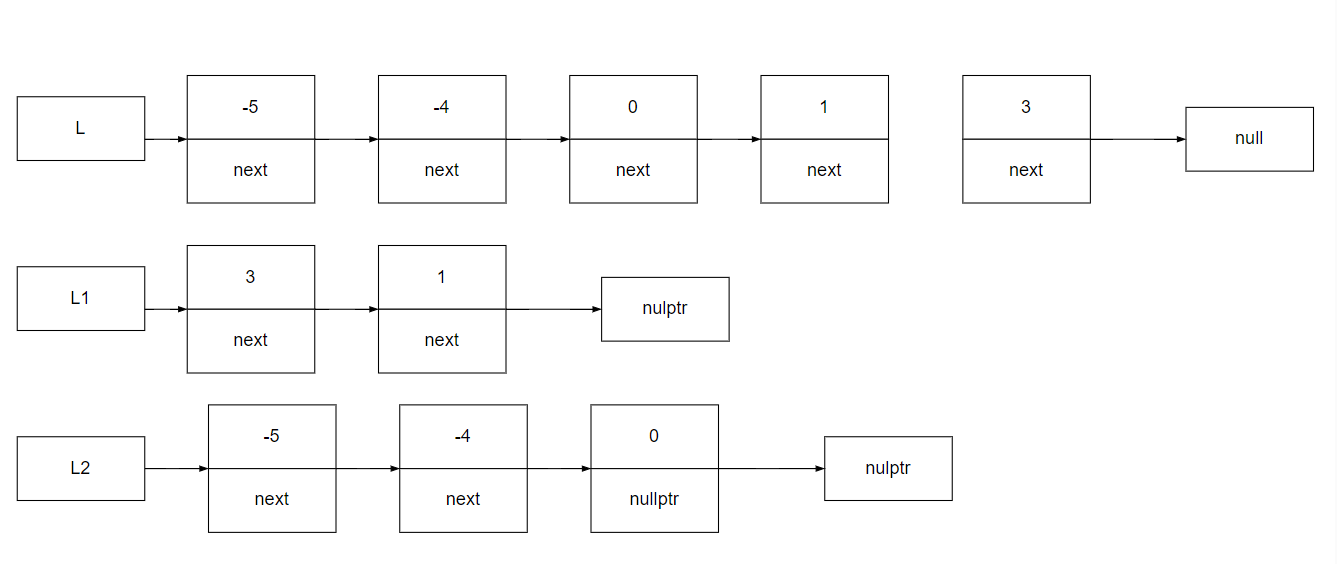
**Before**



**After**



**2.3 Изобразите структуру данных, которая будет использоваться в операциях.**



**2.4 Привести алгоритм выполнения операции**

1. Функция insertFront - Добавление узла в начало списка

Алгоритм:

Создается новый узел newNode, который содержит значение value и указывает своим полем next на текущий первый узел списка head.

Затем head переназначается на newNode, так что newNode становится новым первым узлом в списке.

Эта функция позволяет нам добавлять элементы в начало связанного списка, что удобно для быстрого ввода данных.

2. Функция deleteByKey - Удаление узла по значению

Алгоритм:

Сначала проверяется, содержит ли head значение key, которое нужно удалить:

Если да, то head обновляется, чтобы указывать на следующий узел, затем temp удаляется.

Если нет, то выполняется поиск узла с ключом key путём обхода списка:

Если узел с ключом найден, то prev->next обновляется для пропуска узла temp, и temp удаляется.

Если key не найден, функция завершает выполнение.

Эта функция удаляет узел из списка на основе значения (ключа), указанного пользователем.

3. Функция splitList - Разделение списка L на L1 и L2

Алгоритм:

Выполняется обход каждого узла в списке L.

Если значение узла положительное, узел добавляется в список L1 (список с положительными значениями) с помощью функции insertFront.

Если значение не положительное (отрицательное или 0), узел добавляется в список L2.

Процесс продолжается до конца списка L.

Эта функция делит исходный список на два новых: один содержит положительные значения, а второй — все остальные значения.

4. Функция removeNegatives - Удаление отрицательных элементов из списка L2

Алгоритм:

Выполняется обход списка L2 с проверкой каждого узла:

Если узел содержит отрицательное значение и является первым, то L2 обновляется для пропуска узла, а сам узел удаляется.

Если узел находится в середине или в конце, то обновляется ссылка prev, чтобы пропустить узел с отрицательным значением, и он удаляется.

Если узел не содержит отрицательного значения, prev и temp перемещаются на следующий узел.

Эта функция помогает удалить все отрицательные значения из списка L2.

5. Функция moveMaxToFront - Перемещение максимального элемента в начало списка L1

Алгоритм:

Сначала maxNode указывается на L1 (предполагается, что первый узел — максимальный).

Выполняется обход списка L1 для поиска узла с наибольшим значением:

Если находится узел с большим значением, maxNode и prevMax обновляются.

Если maxNode не является первым узлом, ссылки обновляются так, чтобы maxNode стал первым узлом в списке L1.

Эта функция находит и перемещает максимальное значение в начало списка L1.

6. Функция printList - Вывод списка на экран

Алгоритм:

Выполняется обход списка от head, печатается значение каждого узла, а затем переходит к следующему узлу.

Печатается nullptr, когда список завершается.

Эта функция позволяет вывести значения списка на экран, чтобы пользователь мог видеть результат.

7. Функция inputList - Ввод списка L от пользователя

Алгоритм:

Просит пользователя ввести каждое значение до тех пор, пока не будет введено x.

Преобразует введённую строку в целое число и добавляет его в начало списка L с помощью функции insertFront.

Эта функция позволяет пользователю самостоятельно вводить значения в список L гибким способом.

8. Функция menu - Отображение и выполнение выбранных действий

Алгоритм:

Отображает меню с опциями для пользователя:

1: Отобразить список L.

2: Разделить список L на L1 и L2.

3: Удалить отрицательные элементы из L2.

4: Переместить максимальный элемент в начало L1.

5: Выйти из программы.

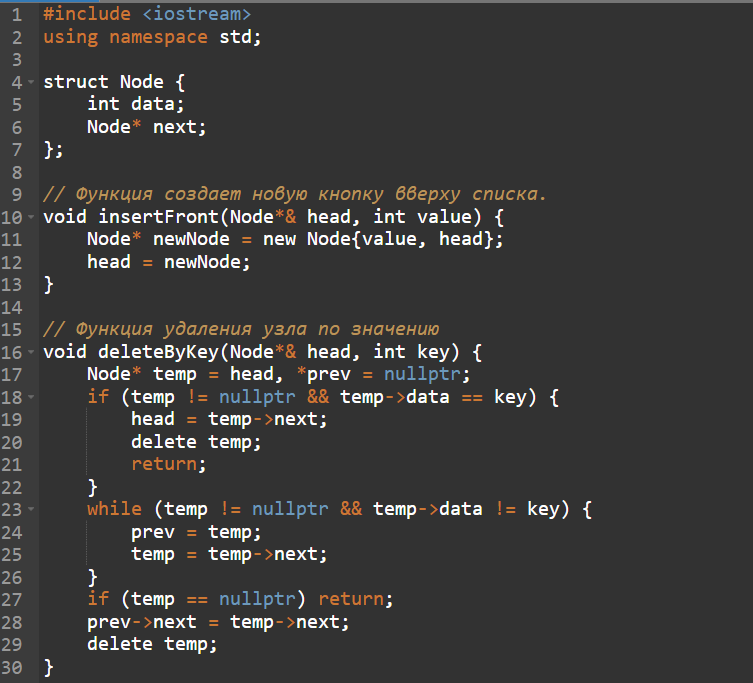
Выполняет функцию, соответствующую выбранному пользователем действию.

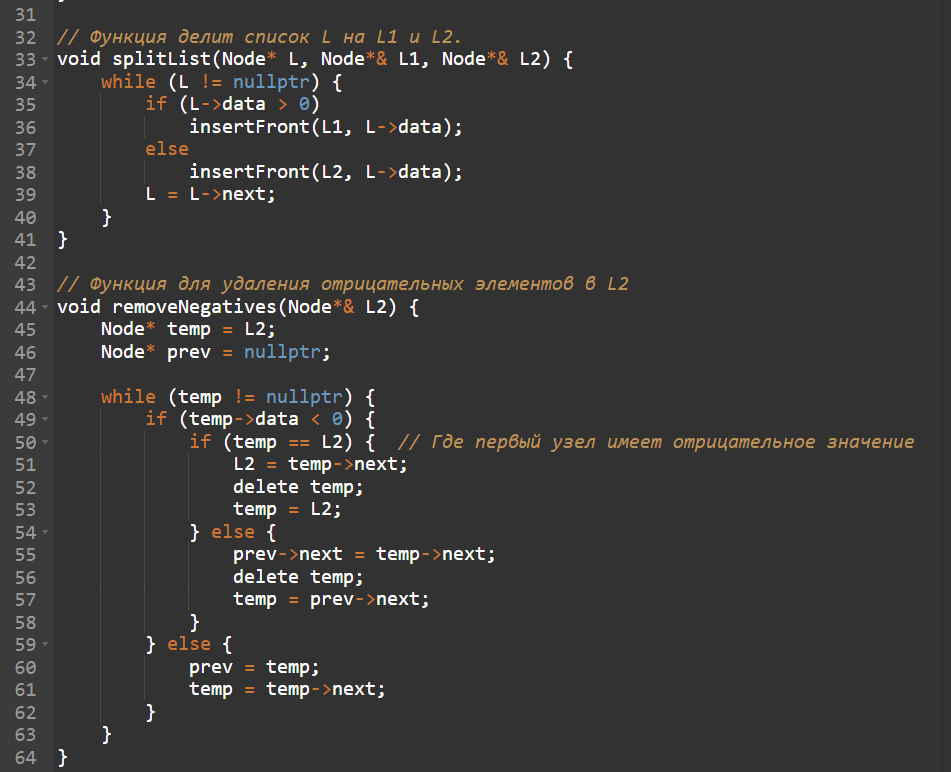
Это основная управляющая функция программы, которая позволяет пользователю выполнять запросы на связанный список.

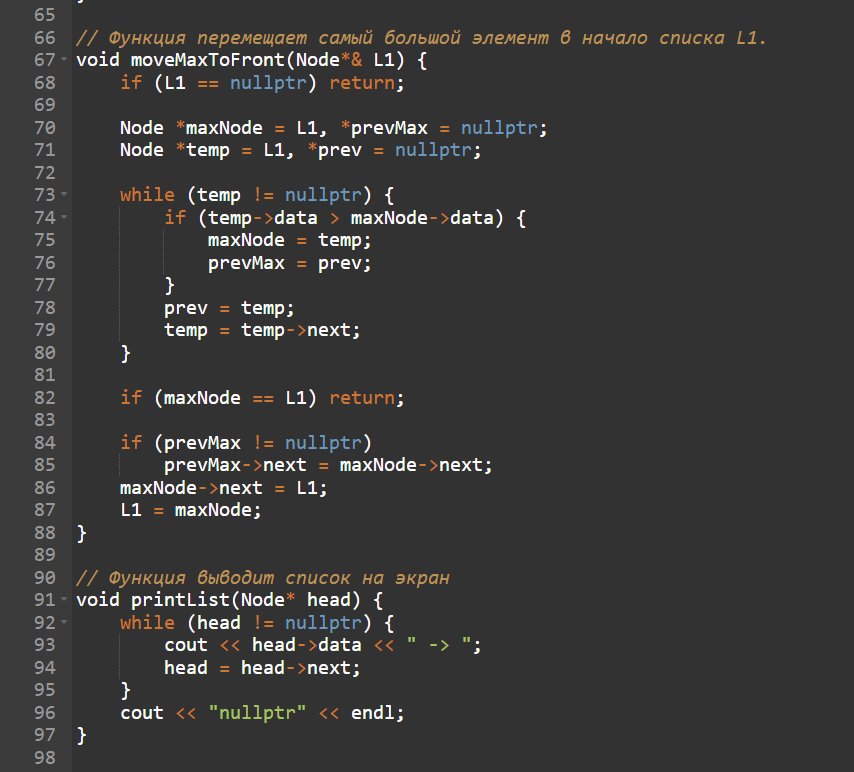
**2.5 Привести таблицу тестов для тестирования каждой операции**

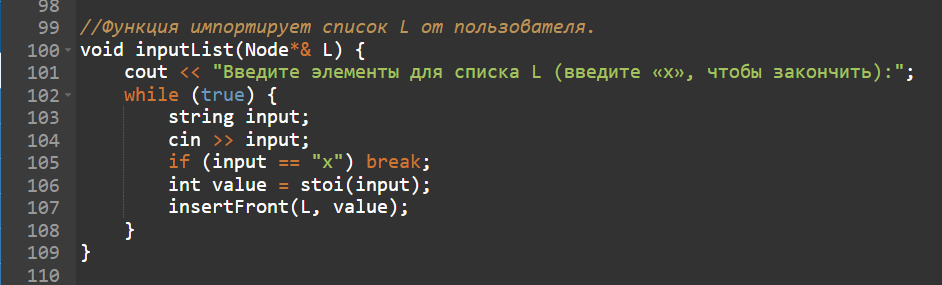
|  |  |  |  |
| --- | --- | --- | --- |
| **Тестовый случай** | **Ввод** | **Ожидаемый вывод** | **Описание** |
| 1. Ввод списка L | 9 10 0 -3 6 8 -7 -5 5 3 x | Список L: 9 -> 10 -> 0 -> -3 -> 6 -> 8 ->-7 -> -5 -> 5 -> 3-> nullptr | Проверка ввода данных и отображения списка. |
| 2. Отображение списка L | 1 | Список L: 3 -> 5 -> -5 -> -7 -> 8 -> 6 -> -3 -> 0 -> 10 -> 9 -> nullptr | Убедитесь, что список L отображается правильно после ввода. |
| 3. Разделение списка на L1 и L2 | 2 | Список Л1 (положительные элементы):9 -> 10 -> 6 -> 8 -> 5 -> 3 -> nullptr  Список L2 (остальные элементы):0 -> -3 -> -7 -> -5 -> nullptr | 3. Разделение списка на L1 и L2 |
| 4. Удаление отрицательных элементов из L2 | 3 | L2 после удаления отрицательных элементов: 0 -> nullptr | Проверка удаления отрицательных элементов из списка L2. |
| 5. Перемещение максимального элемента в начало L1 | 4 | L1 после перемещения самого большого элемента:10 -> 9 -> 6 -> 8 -> 5 -> 3 -> nullptr | Проверка перемещения максимального элемента в начало списка L1. |
| 6. Выход из программы | 5 | Выход из программы. | Проверка функции выхода из программы. |

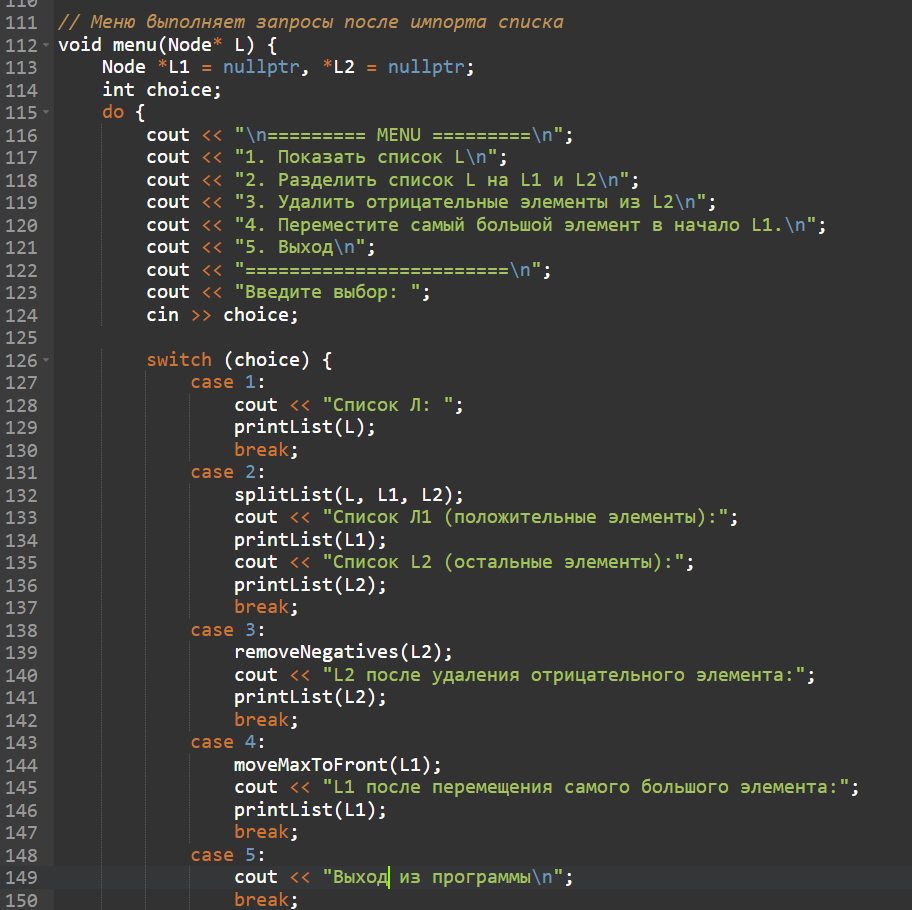
**3. Представить код программы**

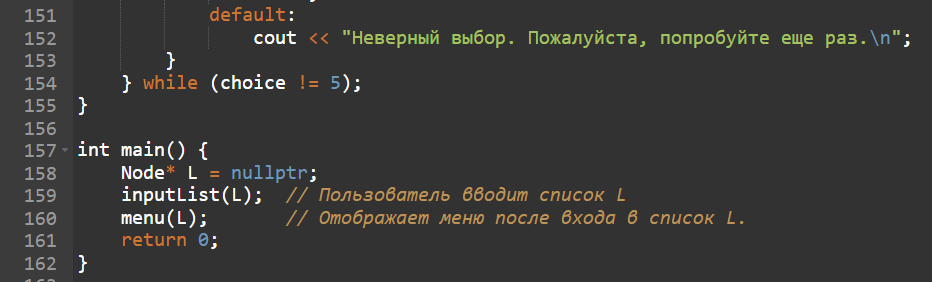




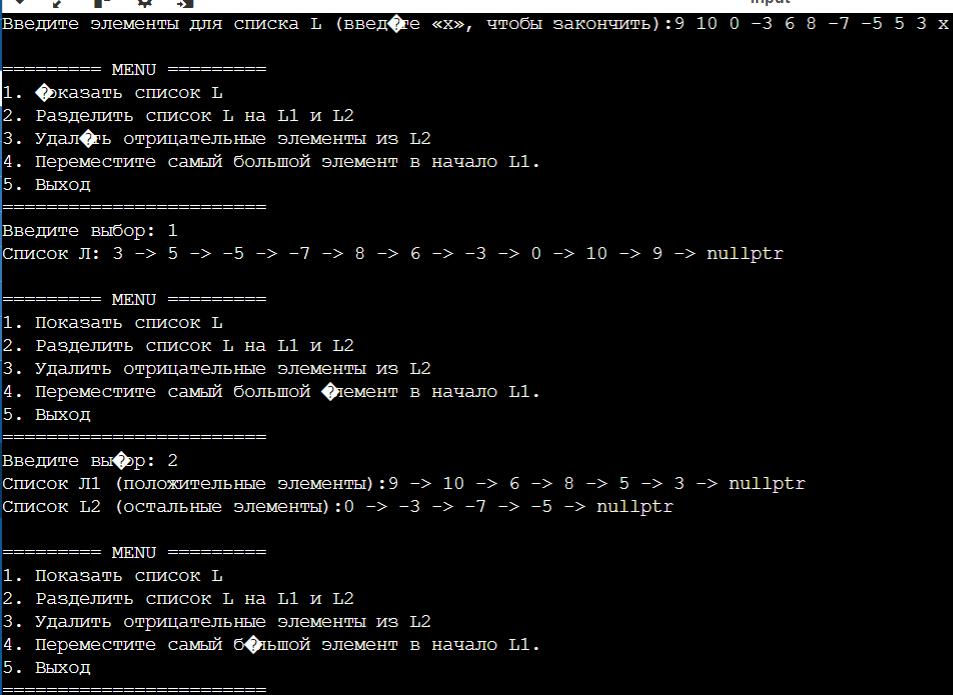


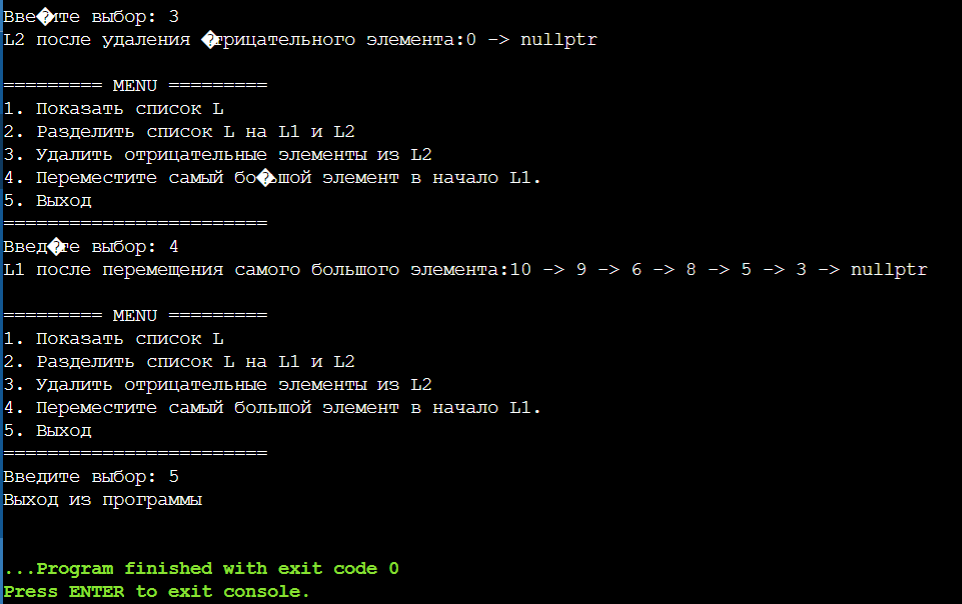






**4. Представить результат тестирования программы: скриншоты выполнения каждой операции.**





**5. Выводы**

Благодаря процессу создания таких функций, как добавление, удаление, разделение, фильтрация и перемещение узлов, это упражнение помогло прояснить структуру данных связанного списка, а также усложнить операции управления и обработки данных.

**6. Список информационных источников, которые были использованы при выполнении задания.**

TopDev, & TopDev. (2023, August 11). *Односвязный список в C++*. TopDev. <https://topdev.vn/blog/danh-sach-lien-ket-don-trong-c/>

Blog T. (1970, January 1). [DSLK đơn]. Удалить средний узел в DSLK. *28Tech Blog*. https://blog.28tech.com.vn/dslk-don-xoa-node-giua-trong-dslk

Kim K. (2023, October 29). *Односвязный список — все подробности.* TEKY - Академия креативных технологий. https://teky.edu.vn/blog/danh-sach-lien-ket-don/