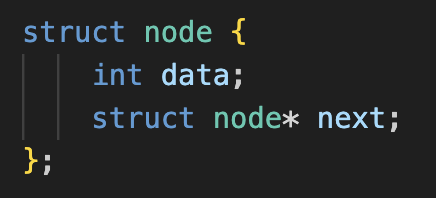
**Отчет курсанта Громова Григория Андреевича группы 22.Б05 о выполнении практического задания на тему «Односвязный список» (пятое задание во втором семестре)**

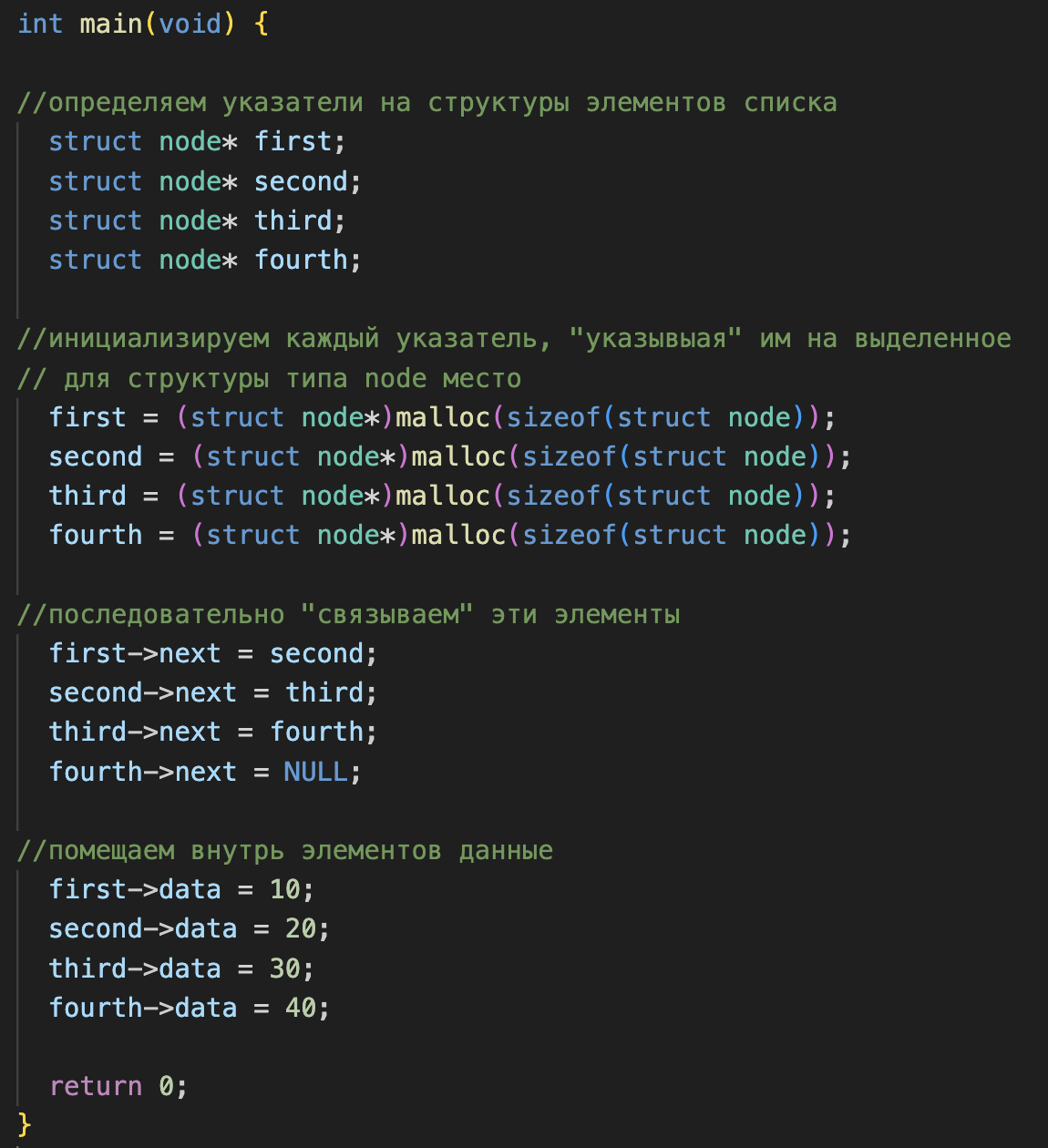
Для хранения данных одной «ячейки» односвязного списка будем использовать структуру node:



data – хранимые данные (например, целое число)

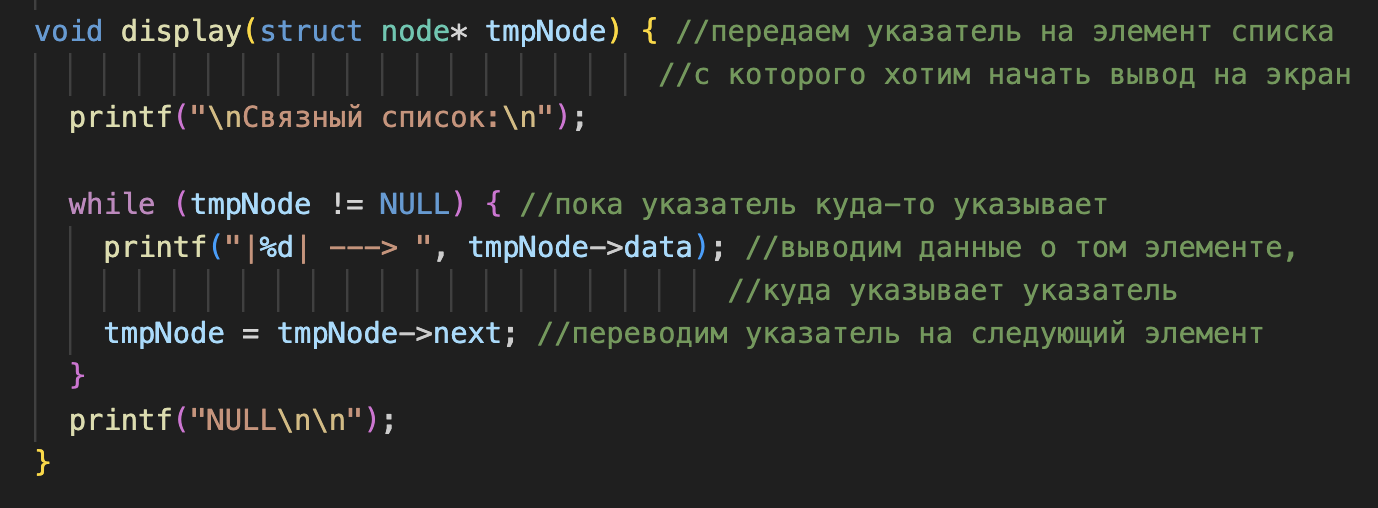
next – указатель на следующий элемент в списке (также на структуру node)

Сначала вручную создадим простенький односвязный список из четырех элементов

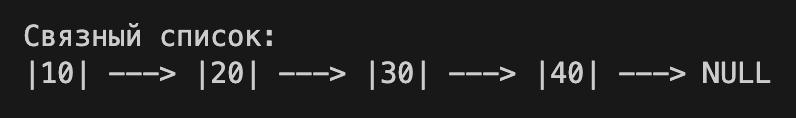


Однако для полноценной работы со связными списками нужно написать несколько важных функций.

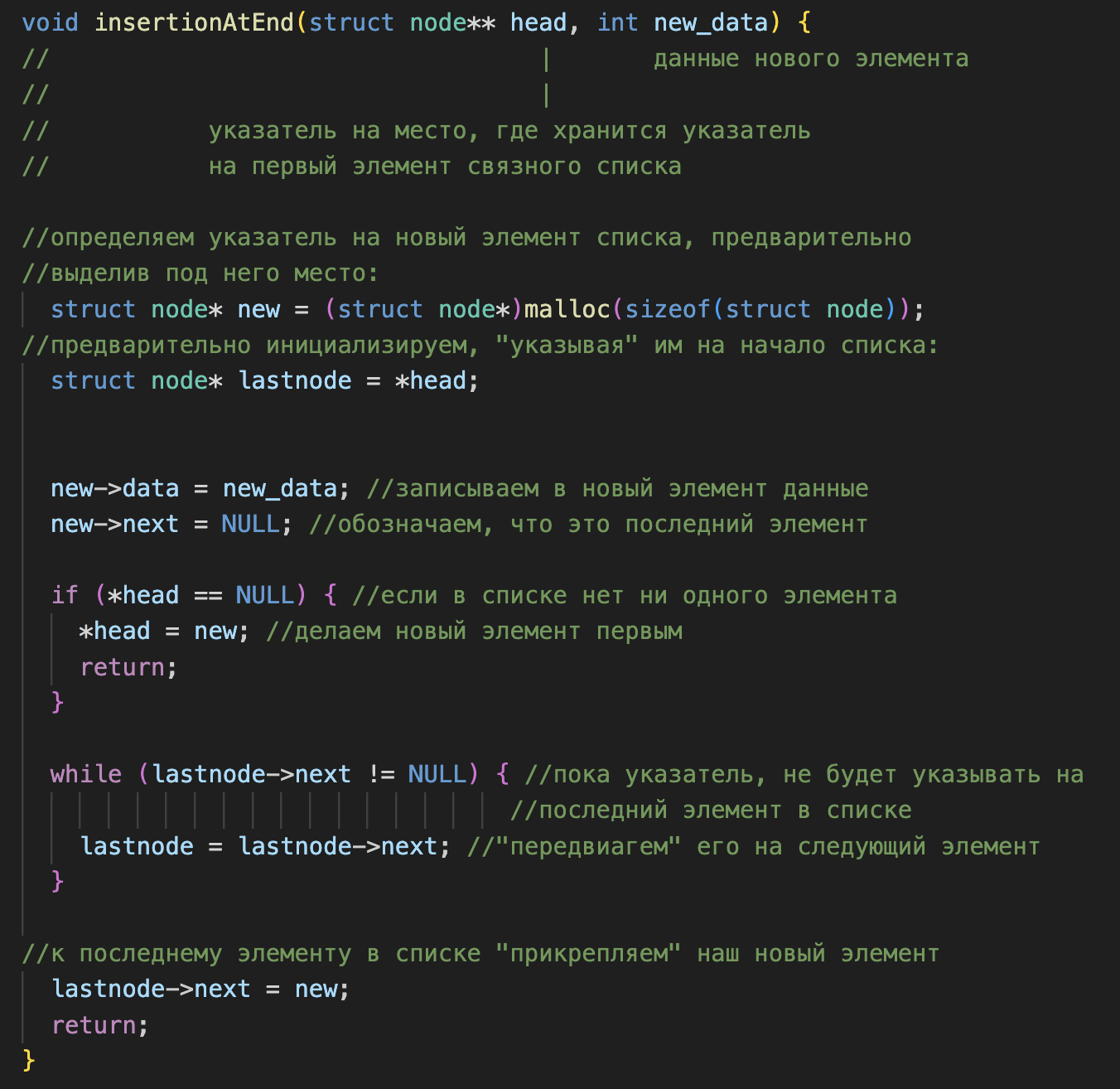
1. Отображение списка



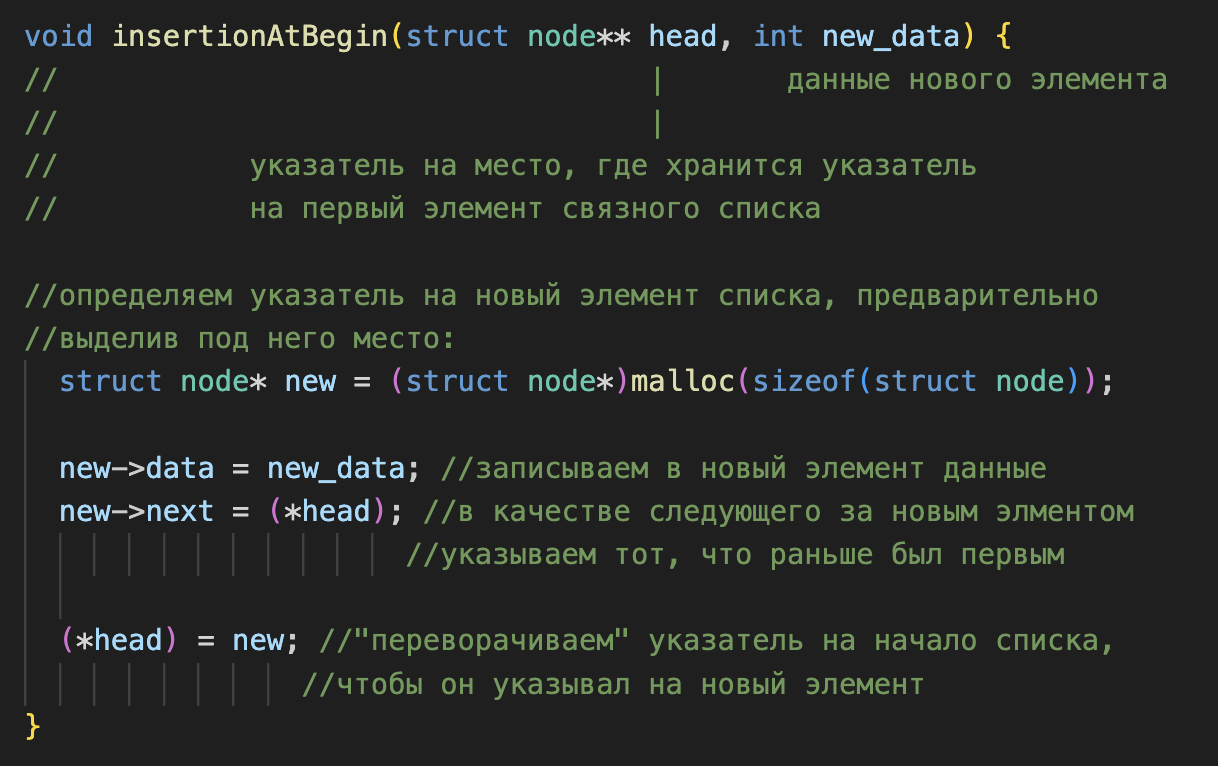
При применении к вышеописанному main(), в терминале появится:



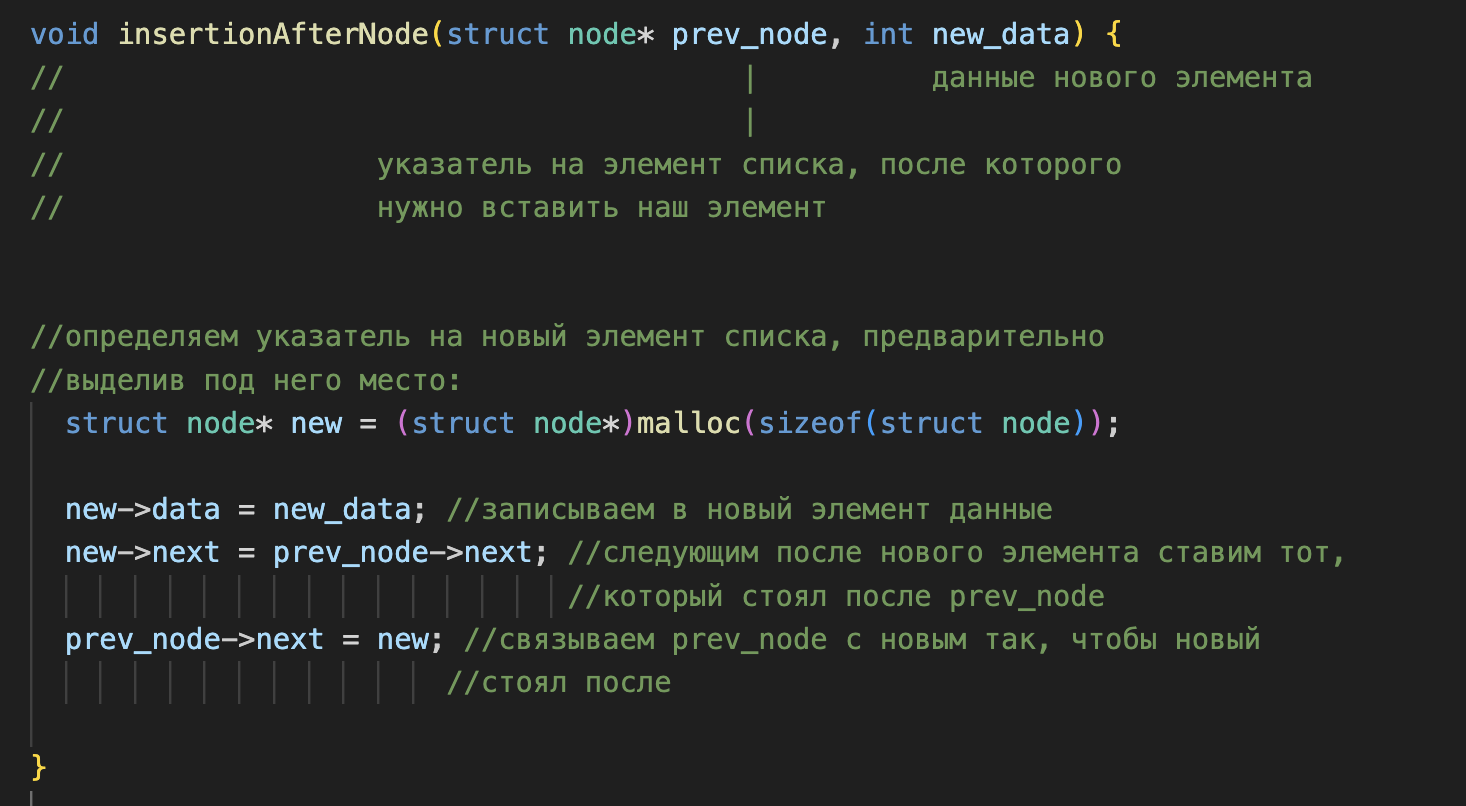
2. Добавление элемента в конец списка



3. Добавление элемента в начало списка

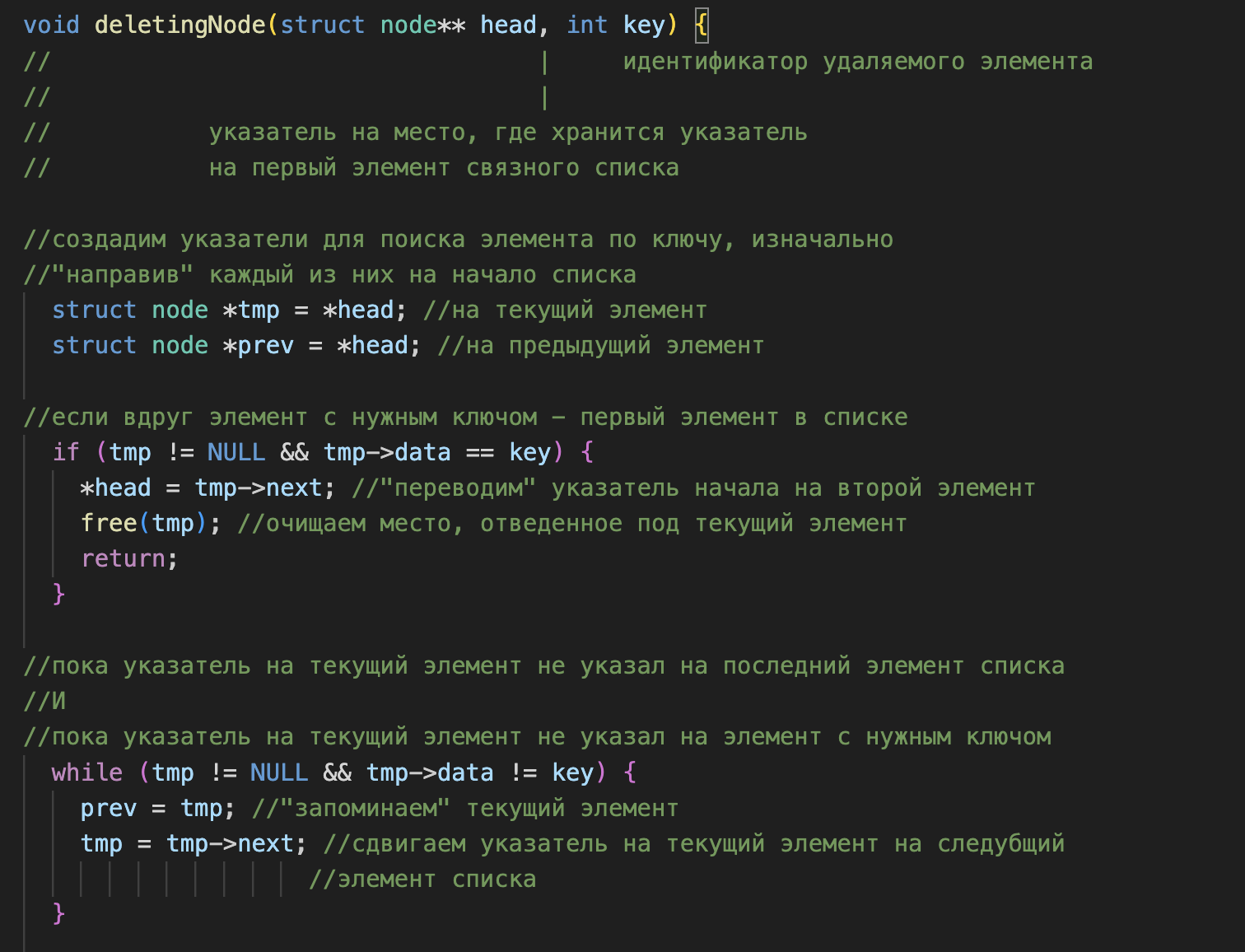


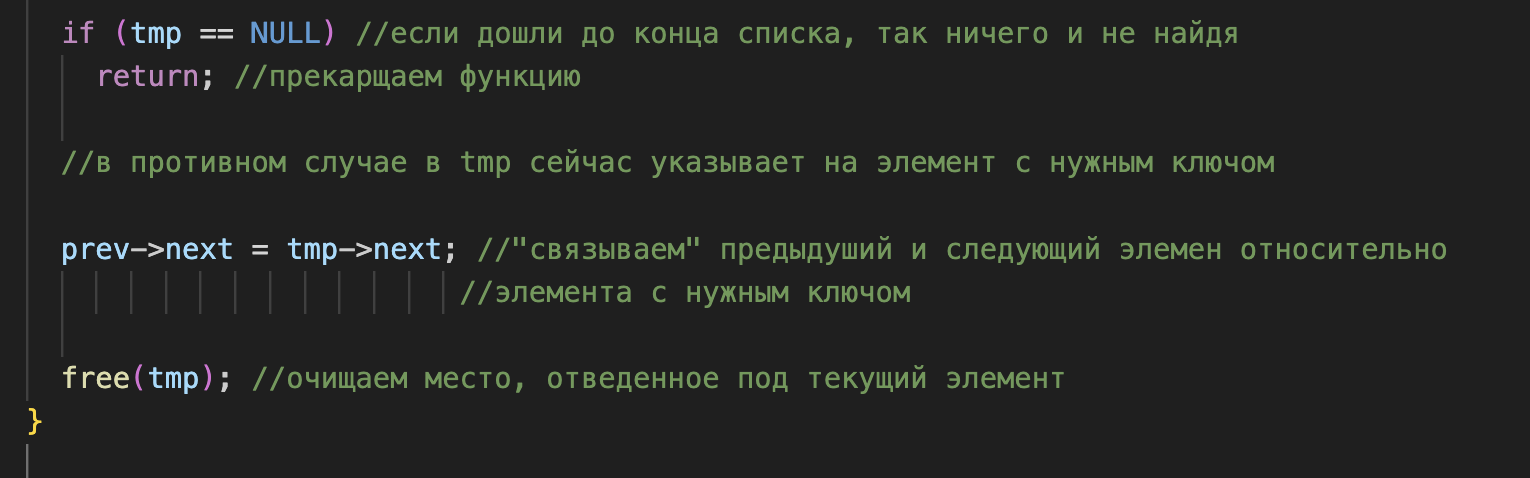
4. Добавление элемента после определенного элемента



5. Удаление элемента

*По-хорошему, при использовании связных списков среди свойств структуры, хранящей данные об элементе списка, должно быть свойство id или нечто подобное (в том числе, чтобы облегчить поиск по списку). Однако в нашем простом пример, такого свойства нет, поэтому в демонстрационных целях для идентификации нужного элемента я буду использовать его «данные» - целое число int.*

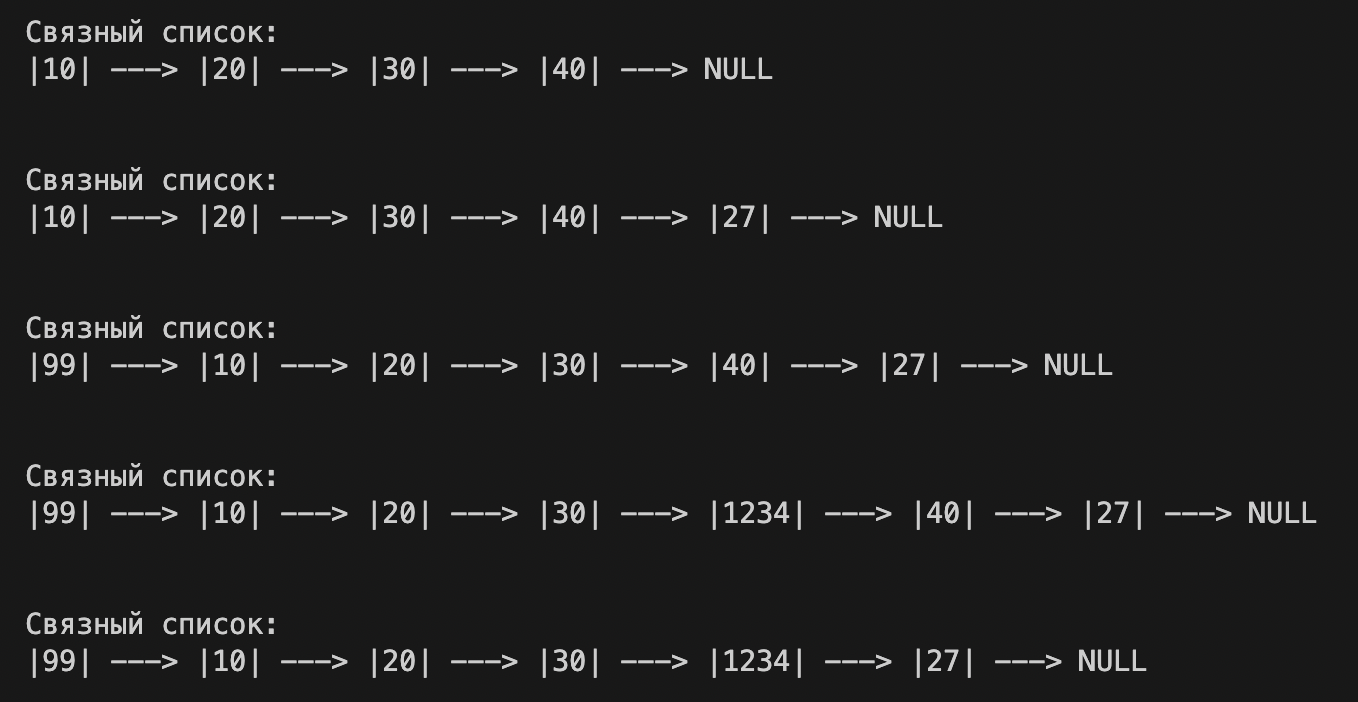




Протестируем наши функции:



В терминале мы увидим:



Все работает корректно, за исключением добавления элемента (1234) после определенного. Мы написали «добавь после третьего», а он добавился после четвертого. Почему так случилось? Дело в том, что данная функция ищет необходимый элемент, ориентируясь на определенный указатель, который мы вручную задали как third. Однако этот порядок сбился, когда мы добавили в начало элемент 99. Fist остался первым, а вот остальные три съехали на один вперед. Во избежании такой проблемы в более серьезный проектах я буду использовать добавление по id, вместо добавления по указателю. Данная реализация была приведена лишь для иллюстрации различных подходов к решению подобных задач.