**Министерство высшего образования и науки Российской Федерации**

**ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)**

**Электротехнический факультет**

**Кафедра Информационные технологии и автоматизированные системы**

**Отчёт по теме**

**«Лабораторная работа № 11»**

**«Информационные динамические структуры»**

**«Очередь»**

**Выполнил:**

**Студент 1 курса**

**Группы ИВТ-22-2б**

**Корючкин Савелий**

**Научный руководитель:**

**Доцент кафедры ИТАС**

**Полякова Ольга Андреевна**

**Пермь 2022**

**Постановка задачи**

Написать программу, в которой создаются динамические структуры и выполнить их обработку.

Записи в линейном списке содержат ключевое поле типа \*string. Сформировать однонаправленный список. Удалить из него К элементов с указанными номерами. Добавить К элементов с указанными номерами.

**Алгоритм решения**

1. Определить узел очереди через структуру struct Node. Узел содержит ключевое слово типа string и указатель на следующий узел.
2. Определить структуру стека struct Line. Структура содержит указатели на начало и конец очереди, конструктор для указателей.
3. Реализовать функцию void push() для добавления элемента в конец очереди. В функции создаётся и заполняется узел очереди, а также обновляется конец очереди.
4. Реализовать функцию для удаления элемента из очереди void pop(). В функции проводится проверка на пустоту очереди, удаляется узел очереди и обновляется её конец.
5. Реализовать функцию void printLine(), которая выводит элементы стека.
6. Реализовать функцию void delete\_key(), с помощью которой происходит удаление заданного ключа и сохраняются элементы изначальной очереди с помощью промежуточных очередей.
7. Реализовать функцию void delete\_elem() для удаления элемента перед элементом с заданным номером.
8. Реализовать функцию void copy\_line1(), с помощью которой происходит копирование элемента из изначальной очереди в промежуточную очередь для выполнения первой части задания.
9. Реализовать функцию void copy\_line2(), с помощью которой происходит копирование элемента из изначальной очереди в промежуточную очередь для выполнения второй части задания.
10. В главной функции int main() реализовать вызов всех внешних функций согласно постановке задачи.

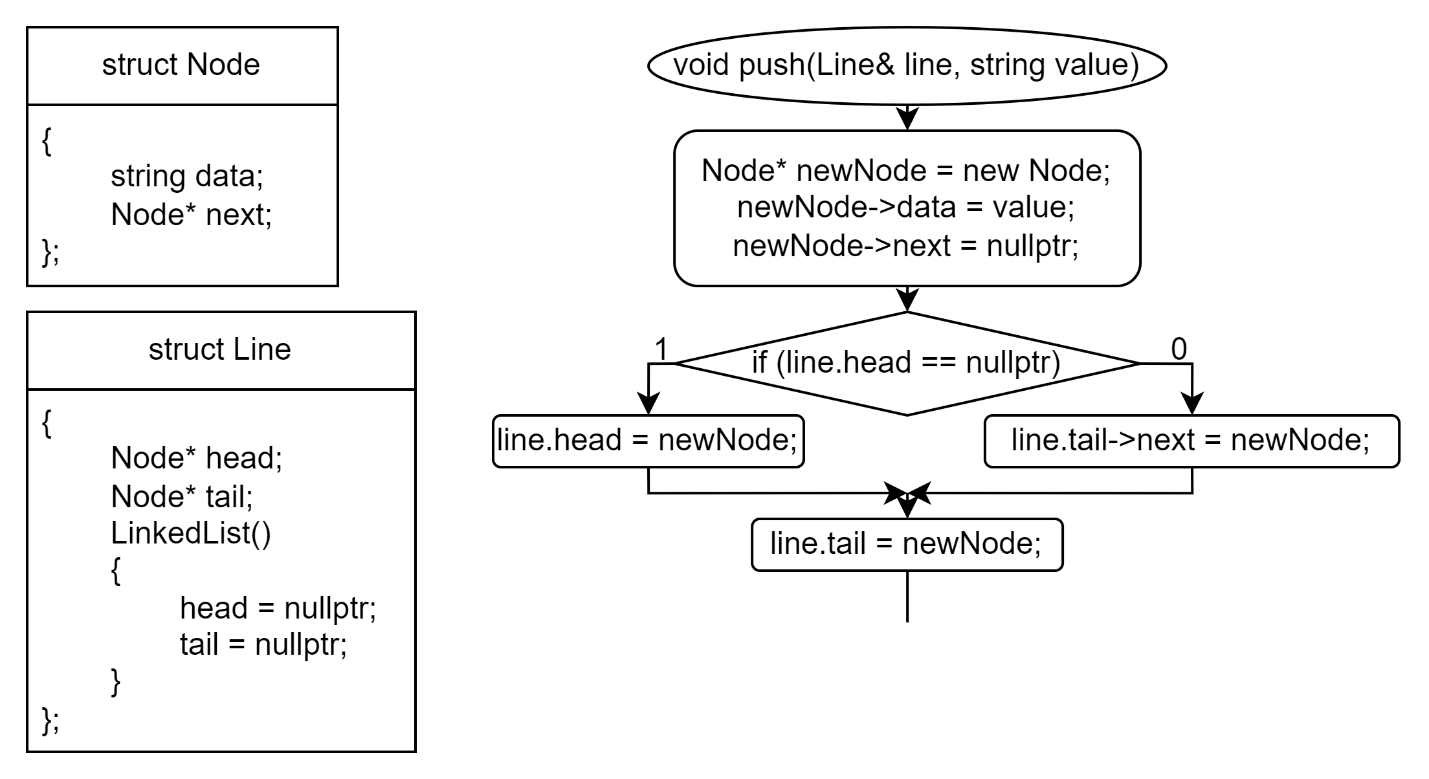


Рис. 1 – Структура узла struct Node, структура списка struct Line, функция void push()

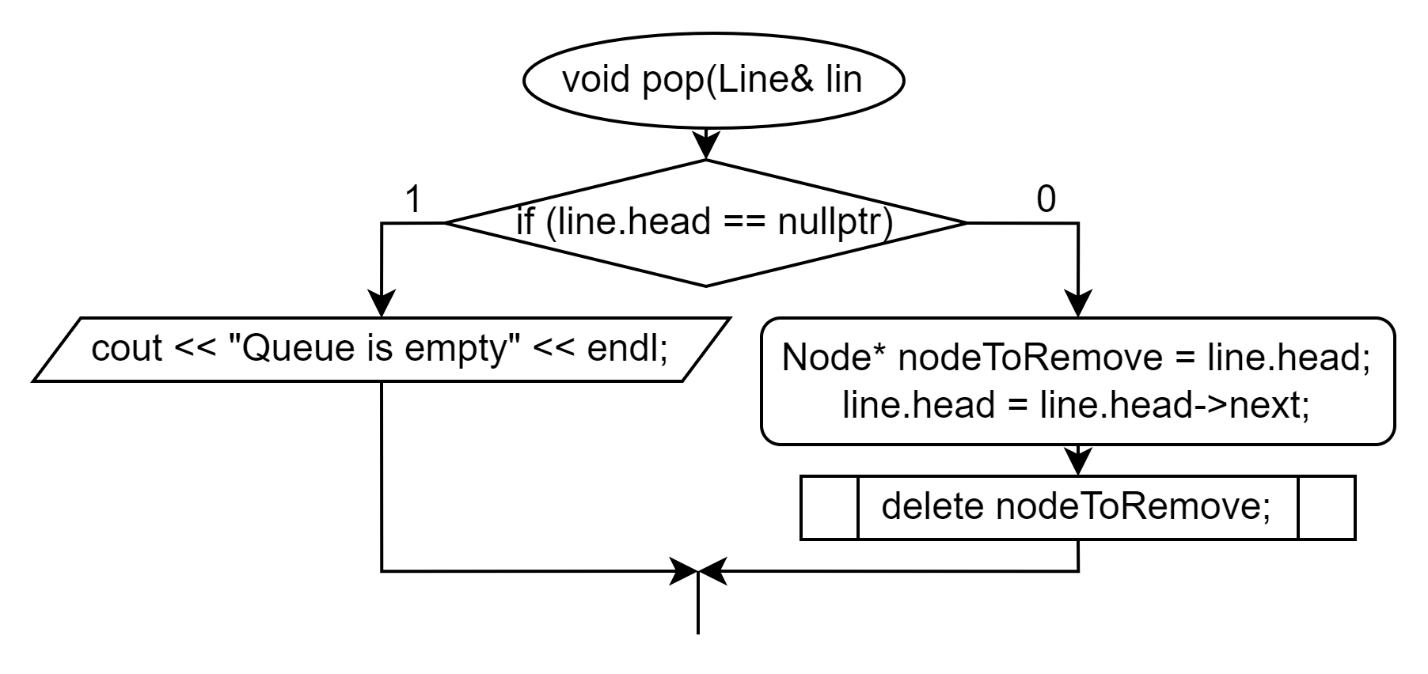


Рис. 2 – Функция void pop()

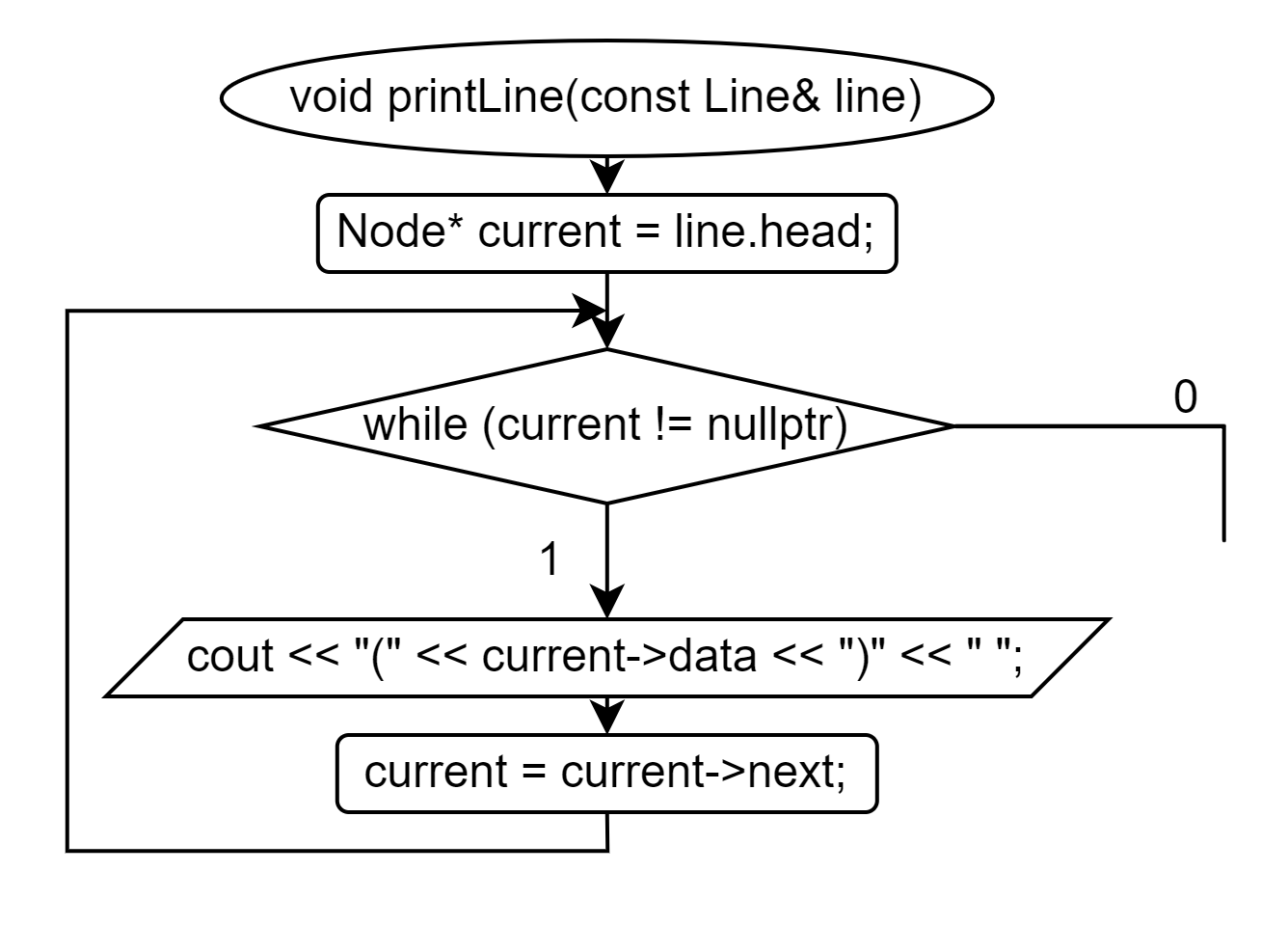


Рис. 3 – Функция void printLine ()

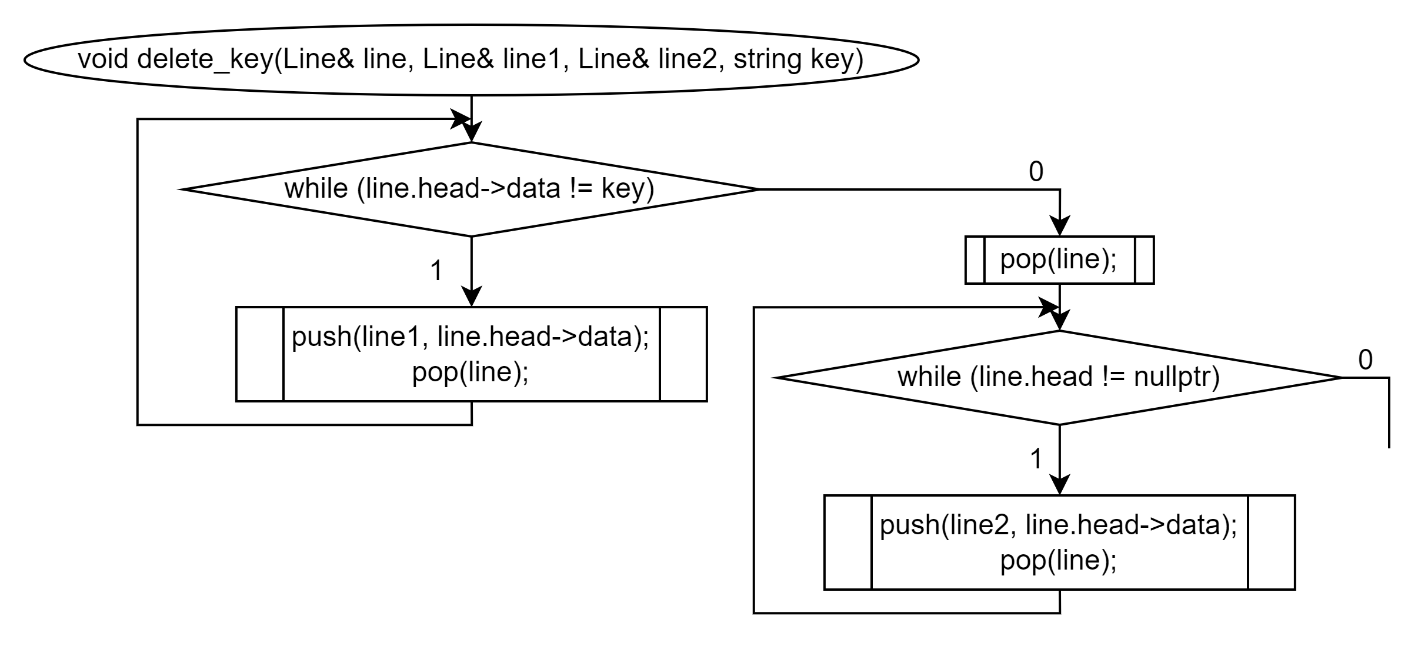


Рис. 4 – Функция void delete\_key()

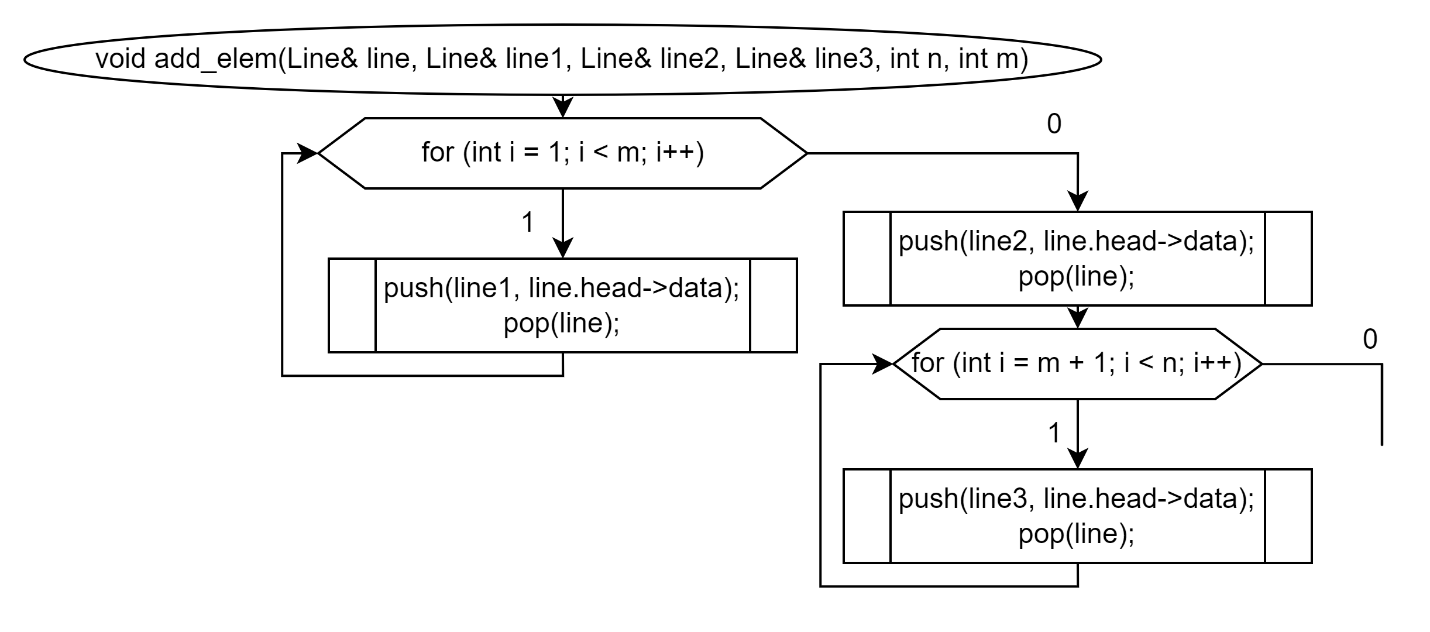


Рис. 5 – Функция void add\_elem ()

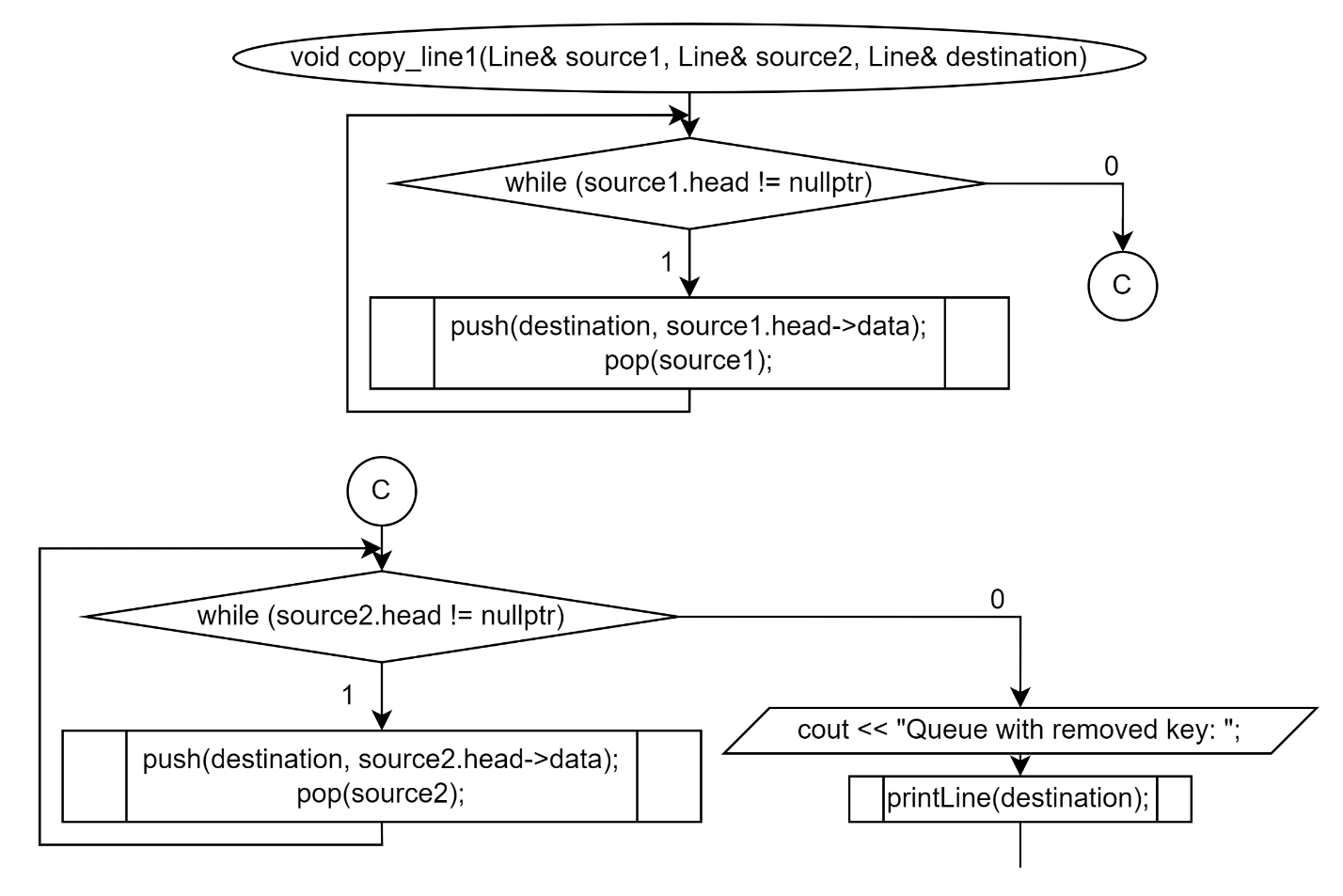


Рис. 6 – Функция void copy\_line1 ()

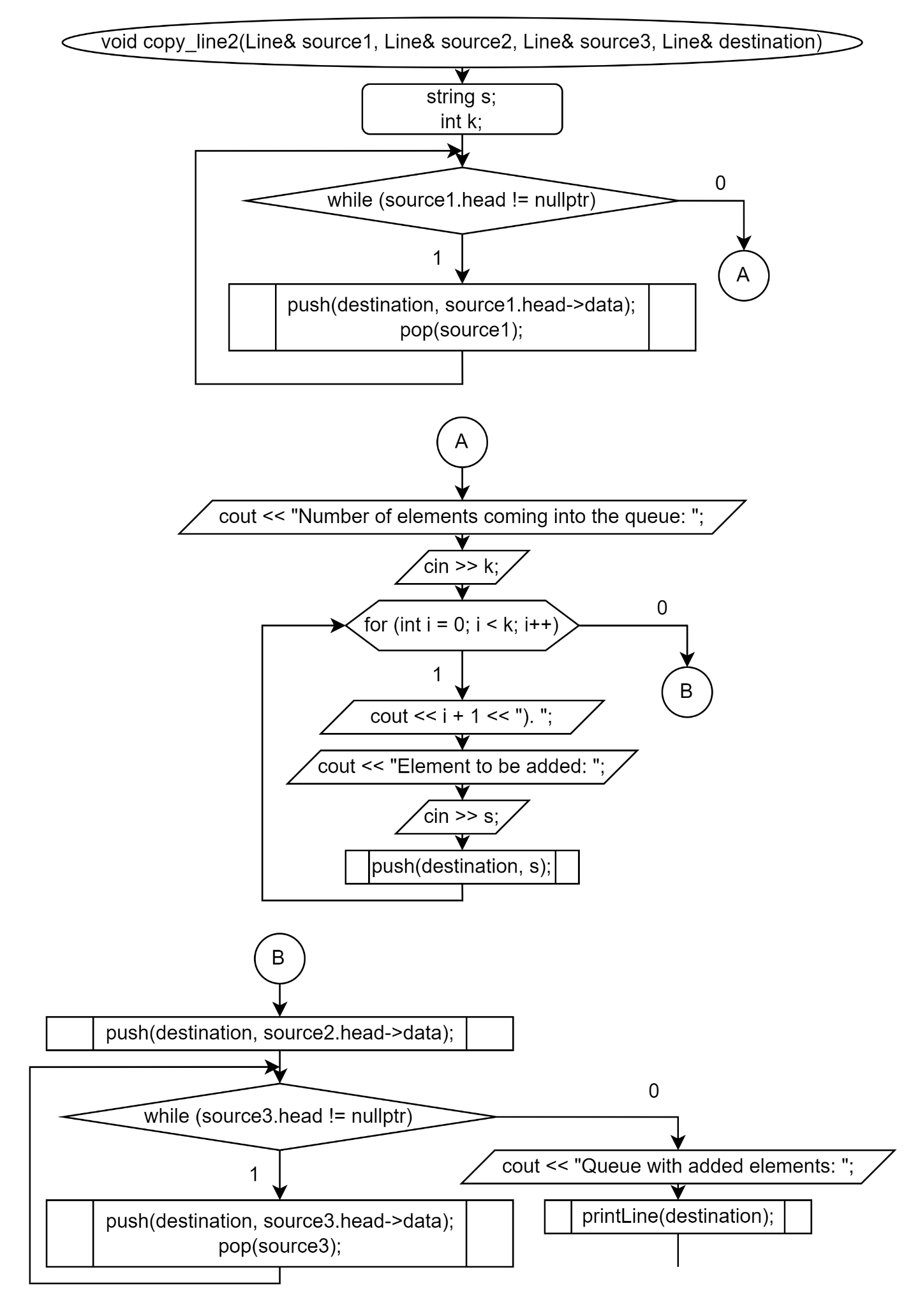


Рис. 7 – Функция void copy\_line2 ()



Рис. 8 – Функция int main()

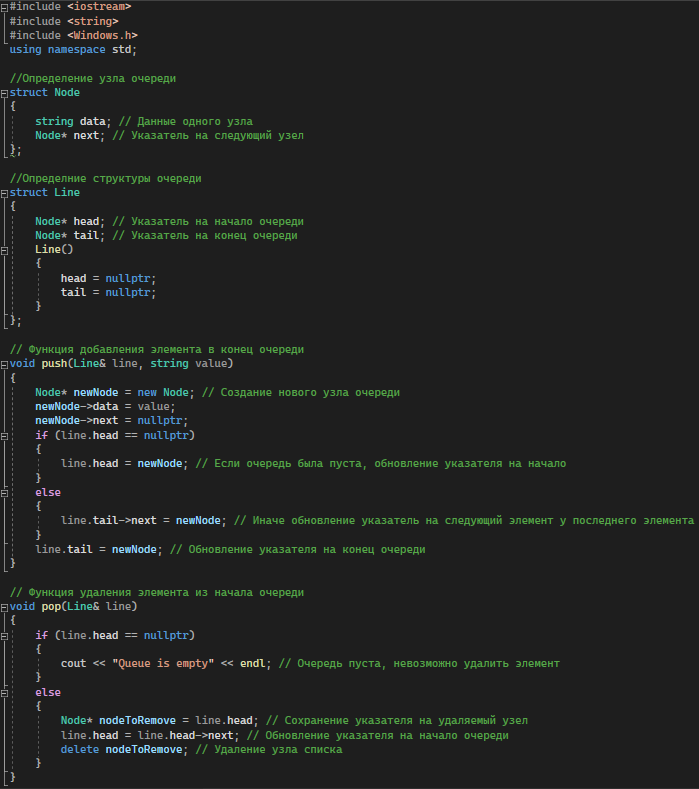


Рис. 9 – Код программы на языке С++

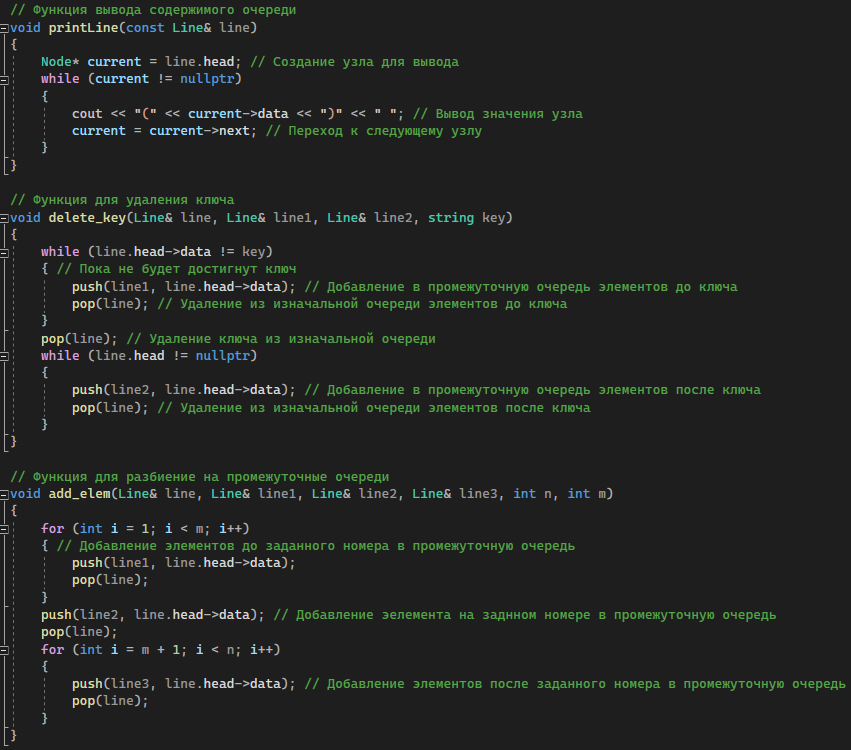


Рис. 10 – Код программы на языке С++

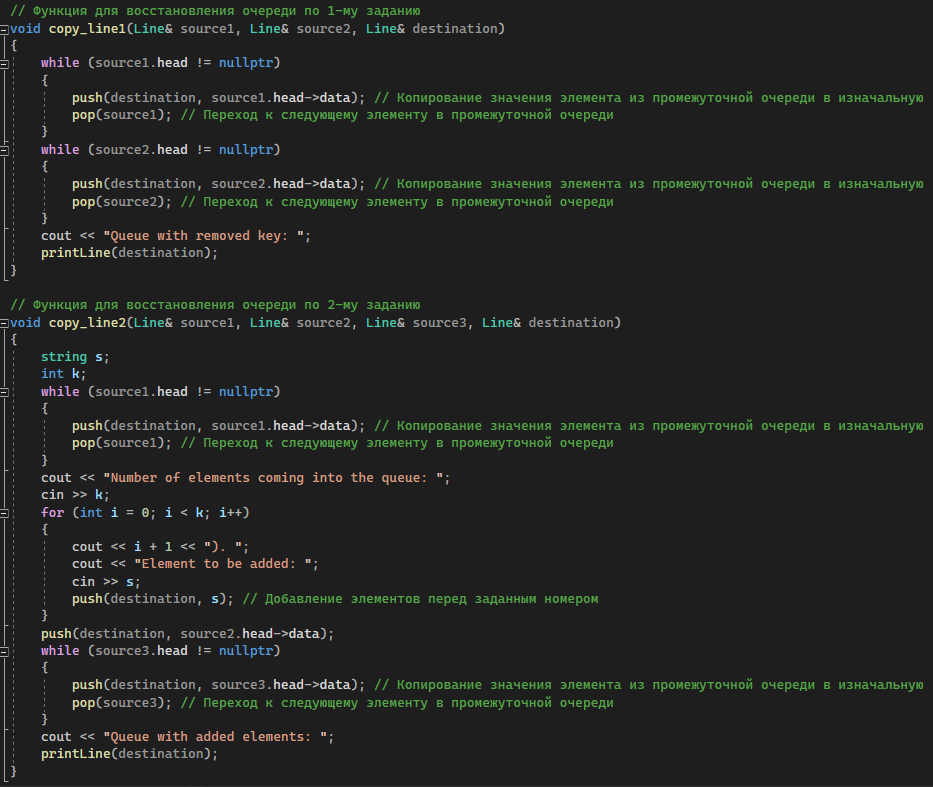


Рис. 11 – Код программы на языке С++

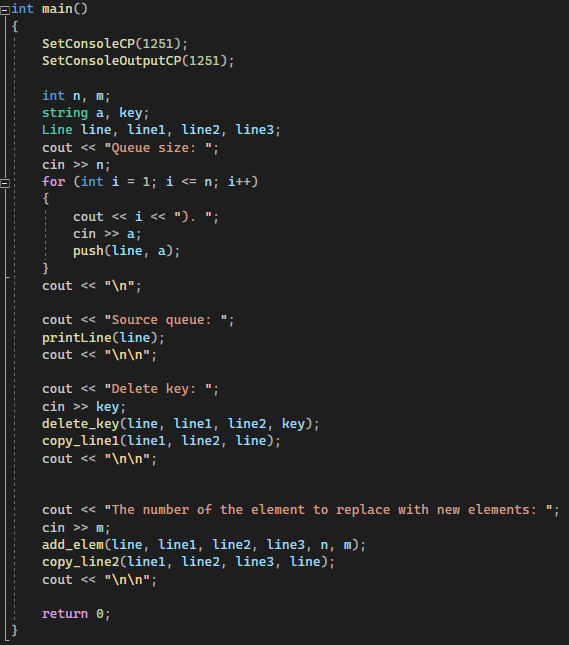


Рис. 12 – Код программы на языке С++

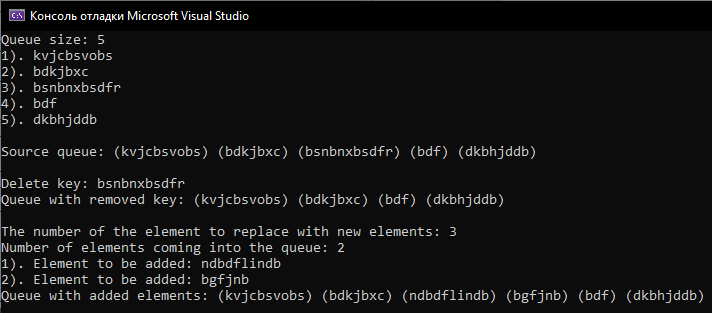


Рис. 13 – Результат работы программы