МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа № 11:**

" Информационные динамические структуры"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Написать программу, в которой создаются динамические структуры и выполнить их обработку для однонаправленного и двунаправленного списков, а так же для стека и очереди. Сформировать линейную структуру, записи в которой содержат ключевое поле типа \*char. Удалить из него К первых элементов. Добавить элемент после элемента, начинающегося с указанного символа.

**Анализ задачи:**

1. Необходимо создать список из элементов типа char. Каждый элемент (узел) содержит информационное поле и поле с ссылкой на следующий элемент (или nullptr).

2. Необходимо добавить несколько узлов списка, поочерёдно перенаправлять ссылки этих элементов.

3. Необходимо задать количество элементов для удаления.

4. Через цикл while удалить к элементов, начиная с начала списка.

5. Необходимо задать информационное поле, после которого нужно поставить новый узел списка.

6. Через цикл for необходимо найти элемент равный заданному и добавить после него новый узел, путём переадресации ссылок.

7. Полученный массив необходимо вывести на экран.

**Код на С++ с комментариями (однонаправленный список):**

#include <iostream>

using namespace std;

struct node// Определение структуры узла списка

{

char data; // Информационное поле узла

node\* next = nullptr; // Указатель на следующий узел, инициализирован как nullptr

};

struct list// Определение структуры списка

{

node\* head = nullptr; // Указатель на начало списка

node\* tail = nullptr; // Указатель на конец списка

};

void printlist(list& l)// Функция печати списка

{

node\* el = l.head;

while (el != nullptr)

{

cout << el->data; // Выводим на экран информационное поле узла

el = el->next; // Переходим к следующему узлу

}

}

void push\_back(list& l, const char& value)// Функция добавления элемента в конец списка (заполнение списка)

{

node\* el = new node(); // Создаем новый узел

el->data = value; // Присваиваем ему значение

if (l.head != nullptr)

{

l.tail->next = el;

l.tail = el;

}

else

{

l.head = el;

l.tail = el;

}

el->next = nullptr;

}

void del(list& l, int k)// Функция для удаления первых k элементов из списка

{

node\* temp;

while (k > 0 && l.head != nullptr) {

temp = l.head;

l.head = l.head->next;

delete temp;

k--;

}

}

void search(list& l, const char& key)// Функция поиска элемента и вставки нового элемента после него

{

node\* now = l.head;

while (now != nullptr)

{

if (now->data == key)

{

char elem;

cout << "Введите элемент, который вы хотите вставить: ";

cin >> elem;

node\* new\_node = new node;

new\_node->data = elem;

new\_node->next = now->next;

now->next = new\_node;

now = new\_node->next;

break;

}

now = now->next;

}

if (now == nullptr)

cout << "Заданного элемента нет в списке!";

}

int main()// Основная функция

{

setlocale(0, ""); // Установка локали

system("chcp 1251"); // Установка кодировки консоли

system("cls"); // Очистка экрана

list l; // Создание списка

char a, b; // Переменные для ввода данных

int n, k;

cout << "Введите размерность списка n: ";

cin >> n;

for (int i = 0; i < n; ++i)

{

cout << "Введите элемент, который вы хотите добавить: ";

cin >> a;

push\_back(l, a);

}

cout << endl << "Получившийся список: " << endl;

printlist(l); // Печать списка

cout << endl << "Введите количество элементов для удаления: ";

cin >> k;

del(l, k); // Удаление К элементов

cout << endl << "Список после удаления К элементов: " << endl;

printlist(l); // Печать списка

cout << endl << "Введите символ после которого хотите вставить новый: ";

cin >> b;

search(l, b); // Поиск и вставка элемента

cout << endl << "Итоговый список: " << endl;

printlist(l); // Печать итогового списка

return 0;

}

**Код на С++ с комментариями (двунаправленный список):**

#include <iostream>

using namespace std;

struct node // Определение структуры узла списка

{

char data; // Информационное поле узла

node\* prev = nullptr; // Указатель на предыдущий узел, инициализирован как nullptr

node\* next = nullptr; // Указатель на следующий узел, инициализирован как nullptr

};

struct list // Определение структуры двунаправленного списка

{

node\* head = nullptr; // Указатель на начало списка

node\* tail = nullptr; // Указатель на конец списка

};

void printlist(list& l) // Функция печати списка

{

node\* el = l.head;

while (el != nullptr)

{

cout << el->data; // Выводим на экран информационное поле узла

el = el->next; // Переходим к следующему узлу

}

}

void push\_back(list& l, const char& value) // Функция добавления элемента в конец списка (заполнение списка)

{

node\* el = new node(); // Создаем новый узел

el->data = value; // Присваиваем ему значение

if (l.head != nullptr)

{

el->prev = l.tail;

l.tail->next = el;

l.tail = el;

}

else

{

l.head = el;

l.tail = el;

}

el->next = nullptr;

}

void del(list& l, int k) // Функция для удаления первых k элементов из списка

{

node\* temp;

while (k > 0 && l.head != nullptr)

{

temp = l.head;

l.head = l.head->next;

l.head->prev = nullptr;

delete temp;

k--;

}

}

void search(list& l, const char& key) // Функция поиска элемента и вставки нового элемента после него

{

node\* now = l.head;

while (now != nullptr)

{

if (now->data == key)

{

char elem;

cout << "Введите элемент, который вы хотите вставить: ";

cin >> elem;

node\* new\_node = new node;

new\_node->data = elem;

new\_node->next = now->next;

new\_node->prev = now;

if (now->next != nullptr)

{

now->next->prev = new\_node;

}

now->next = new\_node;

break;

}

now = now->next;

}

if (now == nullptr)

cout << "Заданного элемента нет в списке!";

}

int main() // Основная функция

{

setlocale(0, ""); // Установка локали

system("chcp 1251"); // Установка кодировки консоли

system("cls"); // Очистка экрана

list l; // Создание списка

char a, b; // Переменные для ввода данных

int n, k;

cout << "Введите размерность списка n: ";

cin >> n;

for (int i = 0; i < n; ++i)

{

cout << "Введите элемент, который вы хотите добавить: ";

cin >> a;

push\_back(l, a);

}

cout << endl << "Получившийся список: " << endl;

printlist(l); // Печать списка

cout << endl << "Введите количество элементов для удаления: ";

cin >> k;

del(l, k); // Удаление К элементов

cout << endl << "Список после удаления К элементов: " << endl;

printlist(l); // Печать списка

cout << endl << "Введите символ после которого хотите вставить новый: ";

cin >> b;

search(l, b); // Поиск и вставка элемента

cout << endl << "Итоговый список: " << endl;

printlist(l); // Печать итогового списка

return 0;

}

**Код на С++ с комментариями (стек):**

#include <iostream> // Подключаем библиотеку для ввода/вывода данных

using namespace std; // Используем пространство имен std

struct Node // Объявляем структуру узла стека

{

char data; // Информационное поле узла

};

struct Stack // Объявляем структуру Стека

{

Node nodes[100]; // Максимальный размер стека

int top = -1; // Инициализируем вершину стека

};

void printStack(Stack& l) // Функция для печати стека

{

for (int i = l.top; i >= 0; i--)

{

cout << l.nodes[i].data; // Выводим информационное поле узла

}

}

void push(Stack& l, const char& value) // Функция для добавления элемента в стек

{

if (l.top < 99)

{

l.top++;

l.nodes[l.top].data = value; // Присваиваем значение элементу

}

else

{

cout << "Стек переполнен, невозможно добавить элемент." << endl;

}

}

void pop(Stack& l) // Функция для удаления верхнего элемента из стека

{

if (l.top >= 0)

{

l.top--;

}

else

{

cout << "Стек пуст, нечего удалять." << endl;

}

}

void searchAndInsert(Stack& l, const char& key) // Функция для поиска элемента и вставки нового элемента после него

{

Stack temp;

while (l.top >= 0)

{

if (l.nodes[l.top].data == key)

{

char elem;

cout << "Введите элемент, который вы хотите вставить: ";

cin >> elem;

push(temp, elem);

}

push(temp, l.nodes[l.top].data);

pop(l);

}

while (temp.top >= 0)

{

push(l, temp.nodes[temp.top].data);

pop(temp);

}

}

int main() // Главная функция программы

{

setlocale(0, ""); // Установка локали

system("chcp 1251"); // Установка кодировки консоли

system("cls"); // Очистка экрана; // Установка локали для работы с русскими символами

Stack l; // Создаем стек

char a, b; // Переменные для ввода данных

int n, k;

cout << "Введите размерность стека n: ";

cin >> n;

for (int i = 0; i < n; ++i)

{

cout << "Введите элемент, который вы хотите добавить: ";

cin >> a;

push(l, a);

}

cout << endl << "Получившийся стек: " << endl;

printStack(l); // Печать стека

cout << endl << "Введите количество элементов для удаления: ";

cin >> k;

for (int i = 0; i < k; ++i)

{

pop(l); // Удаление верхних элементов

}

cout << endl << "Список после удаления k элементов: " << endl;

printStack(l); // Печать стека

cout << endl << "Введите символ после которого хотите вставить новый: ";

cin >> b;

searchAndInsert(l, b); // Поиск и вставка элемента

cout << endl << "Итоговый стек: " << endl;

printStack(l); // Печать итогового стека

return 0; // Завершение программы

}

**Код на С++ с комментариями (очередь):**

#include <iostream>

using namespace std;

struct node// Определение структуры узла очереди

{

char data; // Информационное поле узла

node\* next = nullptr; // Указатель на следующий узел, инициализирован как nullptr

};

struct queue// Определение структуры очереди

{

node\* front = nullptr; // Указатель на начало очереди

node\* rear = nullptr; // Указатель на конец очереди

};

void printqueue(queue& q)// Функция печати очереди

{

node\* el = q.front;

while (el != nullptr)

{

cout << el->data; // Выводим на экран информационное поле узла

el = el->next; // Переходим к следующему узлу

}

}

void push(queue& q, const char& value)// Функция добавления элемента в конец очереди (заполнение очереди)

{

node\* el = new node(); // Создаем новый узел

el->data = value; // Присваиваем ему значение

if (q.rear != nullptr)

{

q.rear->next = el;

q.rear = el;

}

else

{

q.front = el;

q.rear = el;

}

el->next = nullptr;

}

void pop(queue& q, int k)// Функция для удаления первых k элементов из очереди

{

node\* temp;

while (k > 0 && q.front != nullptr) {

temp = q.front;

q.front = q.front->next;

delete temp;

k--;

}

}

void search(queue& q, const char& key)// Функция поиска элемента и вставки нового элемента после него

{

node\* now = q.front;

while (now != nullptr)

{

if (now->data == key)

{

char elem;

cout << "Введите элемент, который вы хотите вставить: ";

cin >> elem;

node\* new\_node = new node;

new\_node->data = elem;

new\_node->next = now->next;

now->next = new\_node;

now = new\_node->next;

break;

}

now = now->next;

}

if (now == nullptr)

cout << "Заданного элемента нет в очереди!";

}

int main()// Основная функция

{

setlocale(0, ""); // Установка локали

system("chcp 1251"); // Установка кодировки консоли

system("cls"); // Очистка экрана

queue q; // Создание очереди

char a, b; // Переменные для ввода данных

int n, k;

cout << "Введите размерность очереди n: ";

cin >> n;

for (int i = 0; i < n; ++i)

{

cout << "Введите элемент, который вы хотите добавить: ";

cin >> a;

push(q, a);

}

cout << endl << "Получившаяся очередь: " << endl;

printqueue(q); // Печать очереди

cout << endl << "Введите количество элементов для удаления: ";

cin >> k;

pop(q, k); // Удаление К элементов

cout << endl << "Очередь после удаления К элементов: " << endl;

printqueue(q); // Печать очереди

cout << endl << "Введите символ после которого хотите вставить новый: ";

cin >> b;

search(q, b); // Поиск и вставка элемента

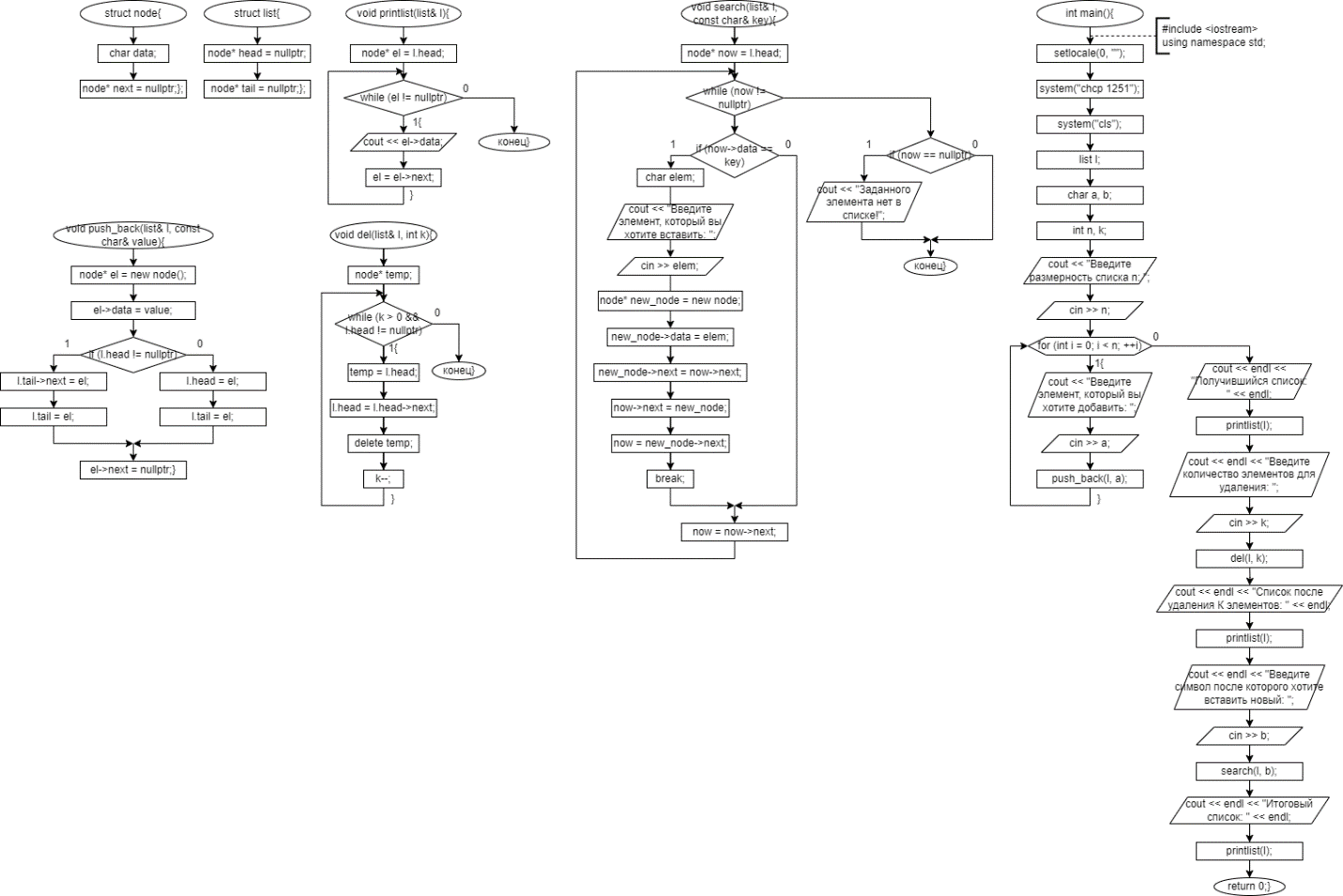
cout << endl << "Итоговая очередь: " << endl;

printqueue(q); // Печать итоговой очереди

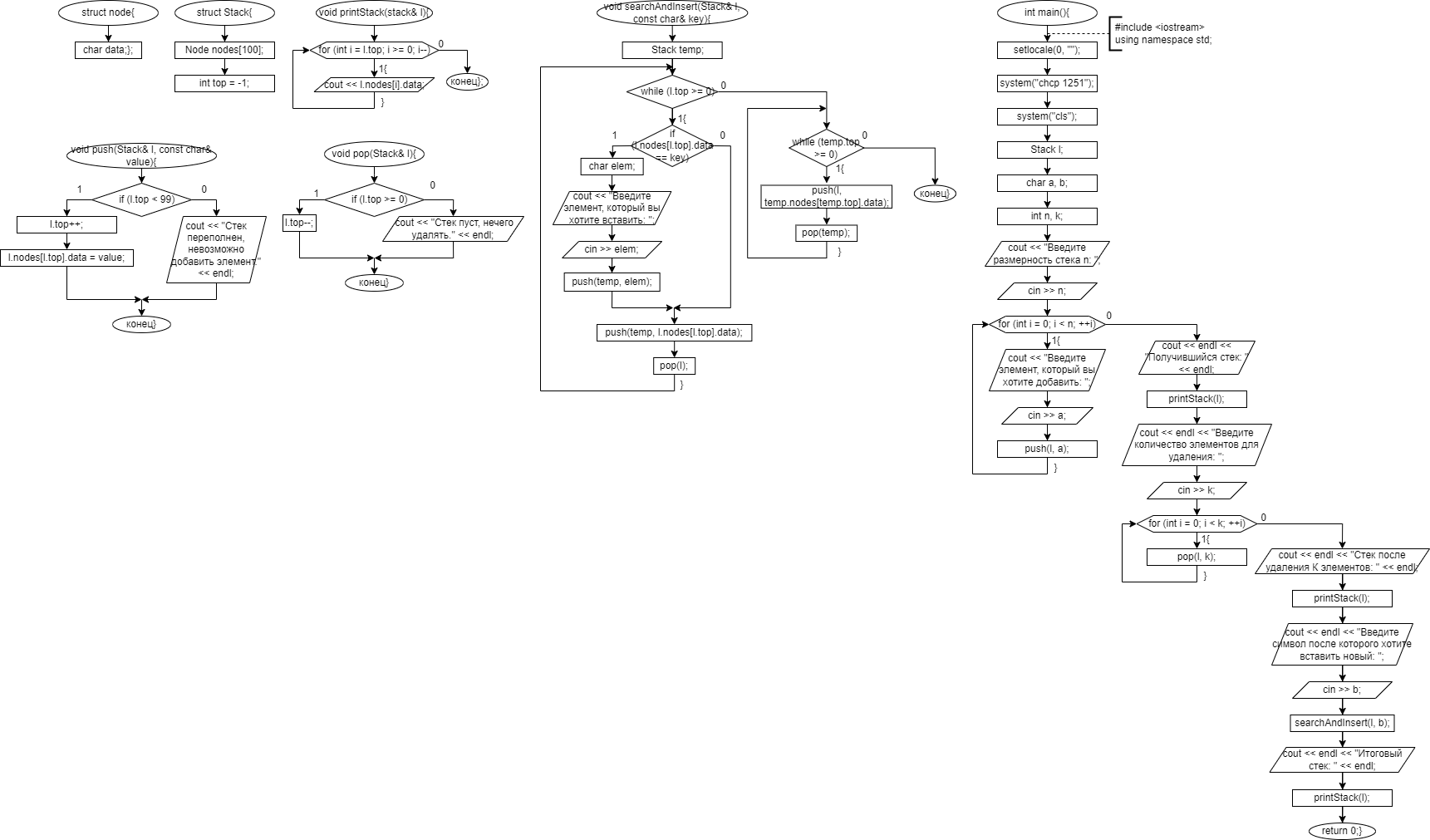
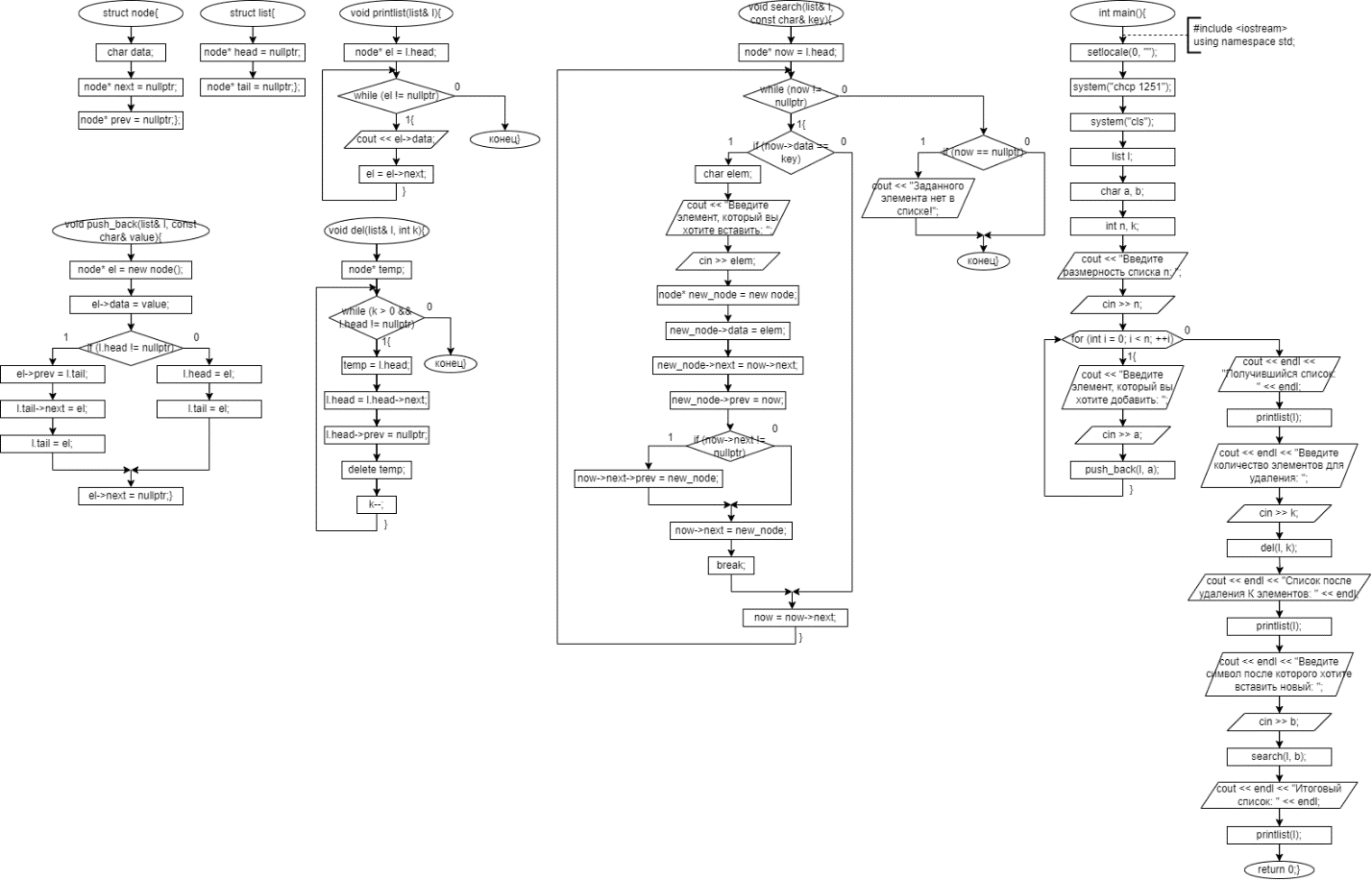
return 0;

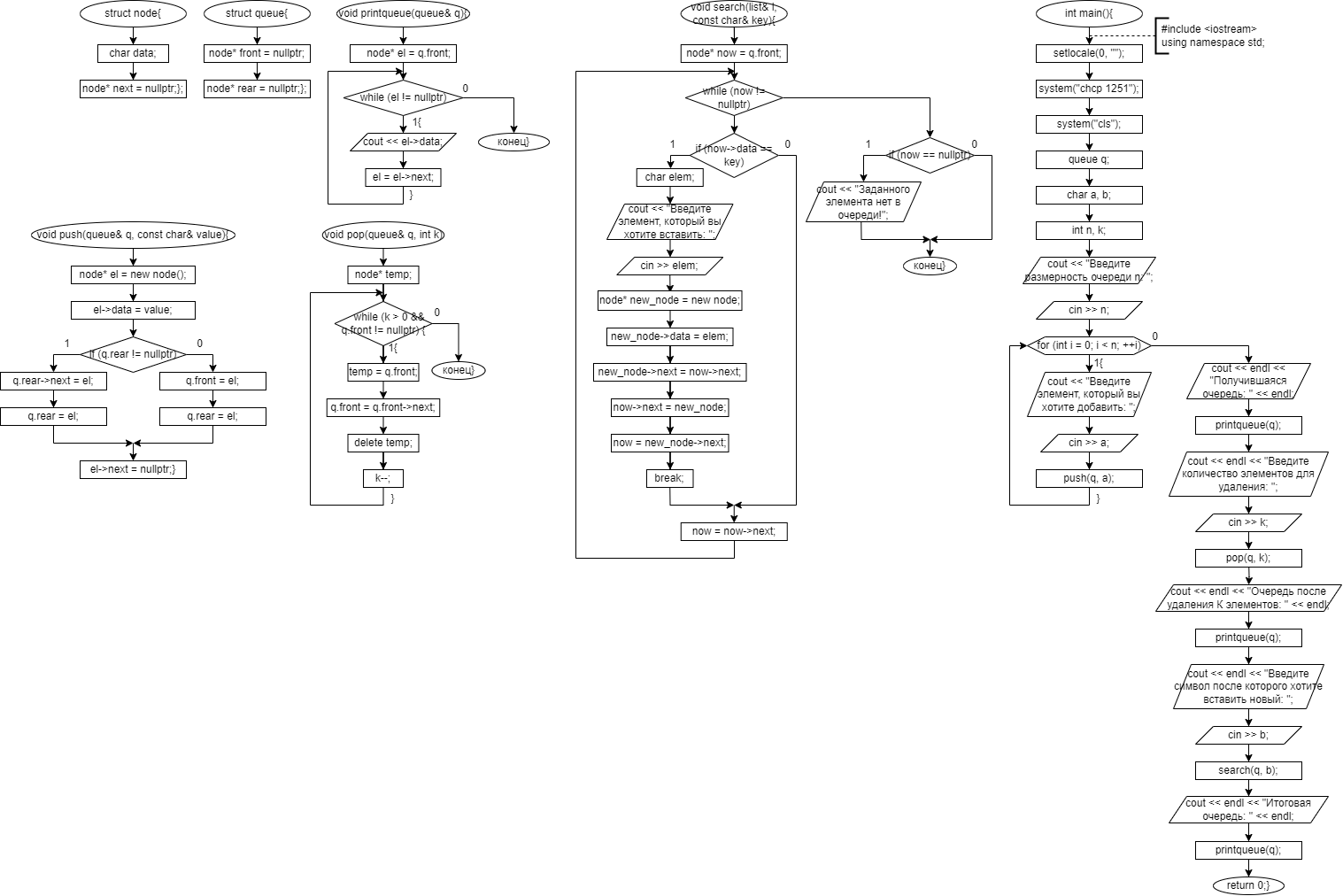
}

**Блок схема:**

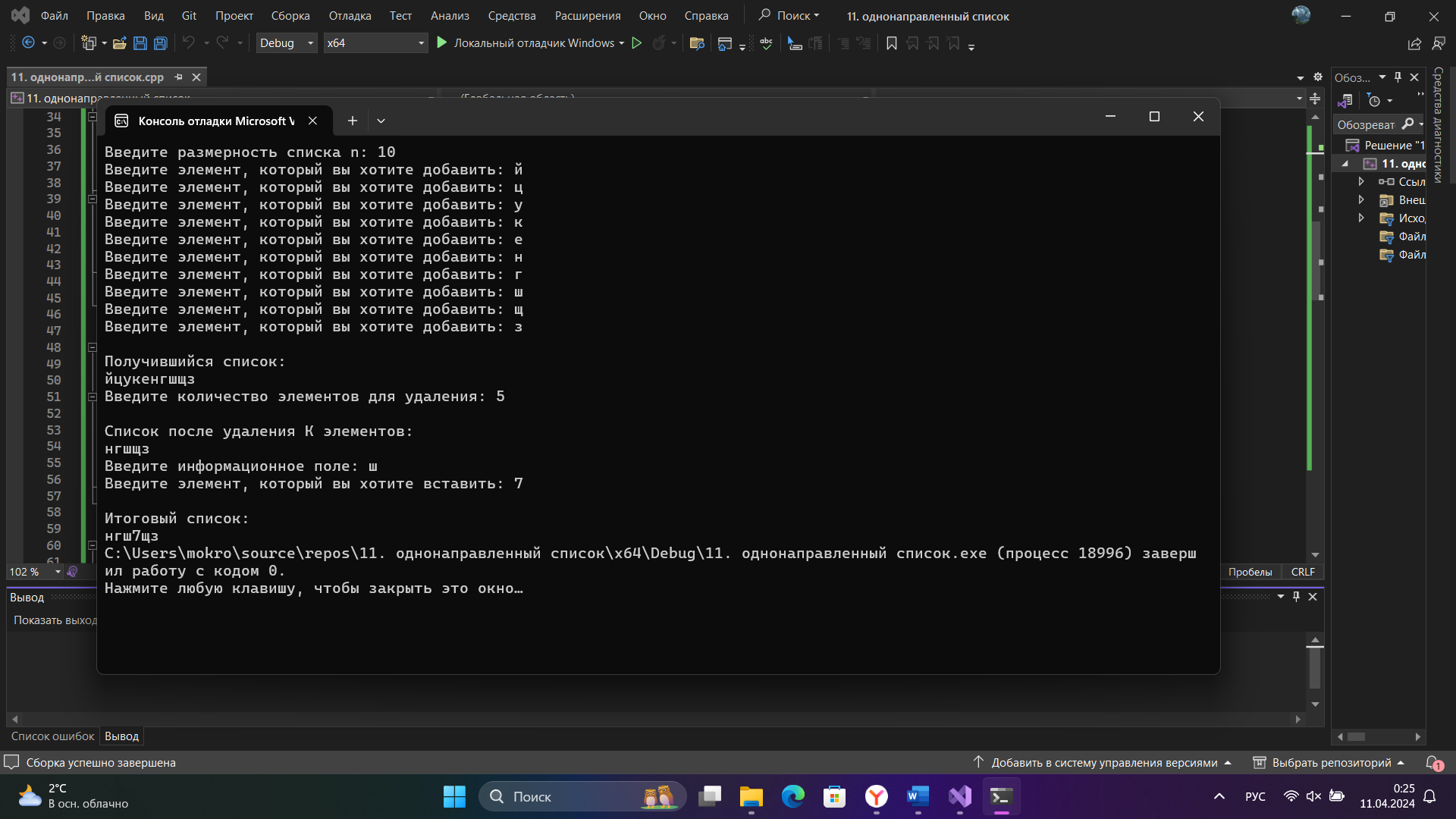
**Однонаправленный список:**

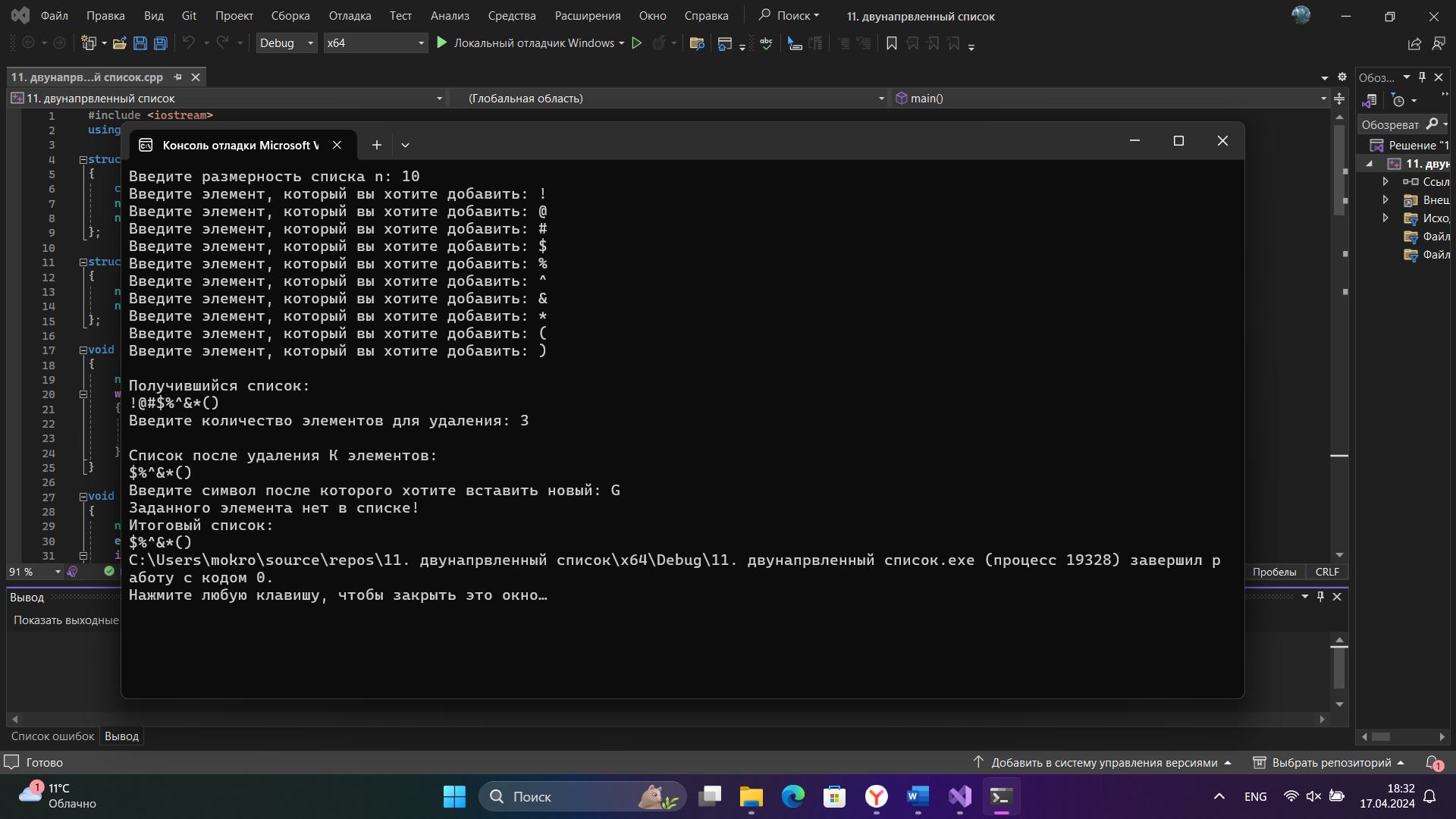
**Двунаправленный список:**

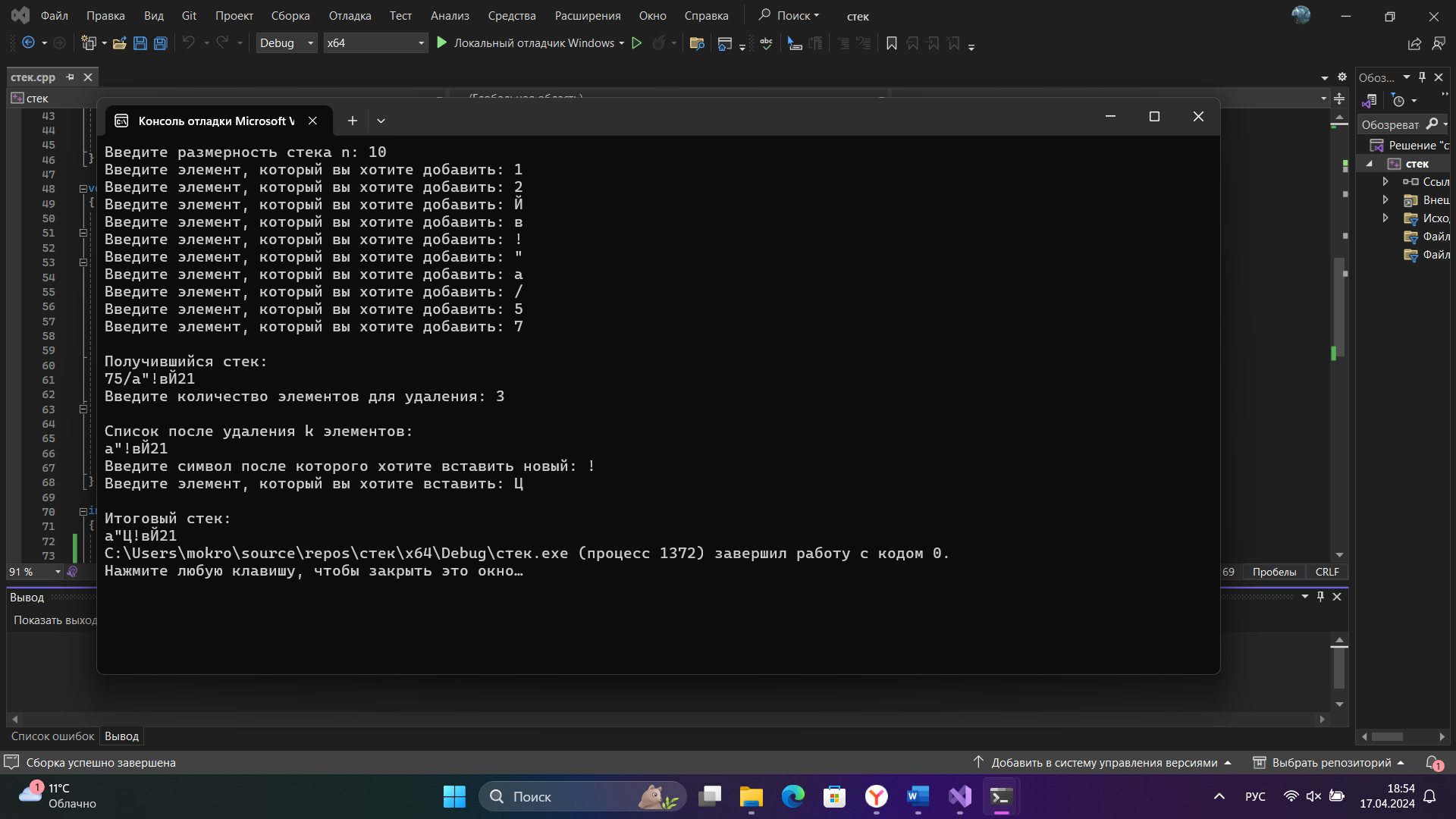
**Стек:**

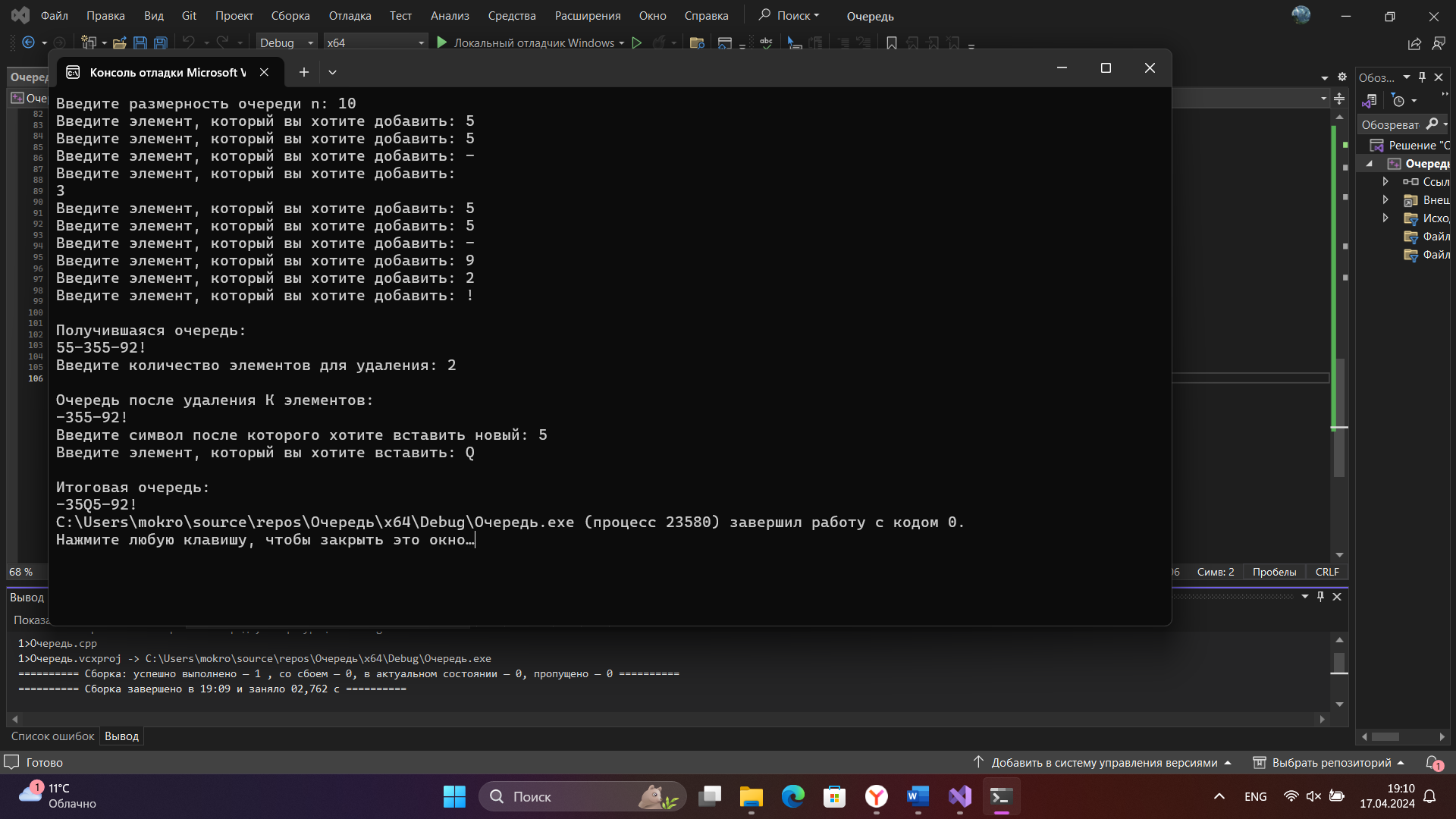
**Очередь:**

**Скриншот работы программы:**









**Вывод:**

Программа успешно выполняет поставленную задачу.

**Скриншот с GitHab:**