Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

Выполнил:   
студент группы РИС-23-1б   
Молодых Никита Андреевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь, 2024 г.

**«Стеки»**

**Условие:**

Написать программу, в которой создаются динамические

структуры и выполнить их обработку в соответствии со

своим вариантом.

Записи в линейном списке содержат ключевое поле типа

\*char(строка символов). Сформировать двунаправленный

список. Удалить К элементов из конца списка. Добавить

элемент после элемента с заданным ключом.

**Анализ задачи:**

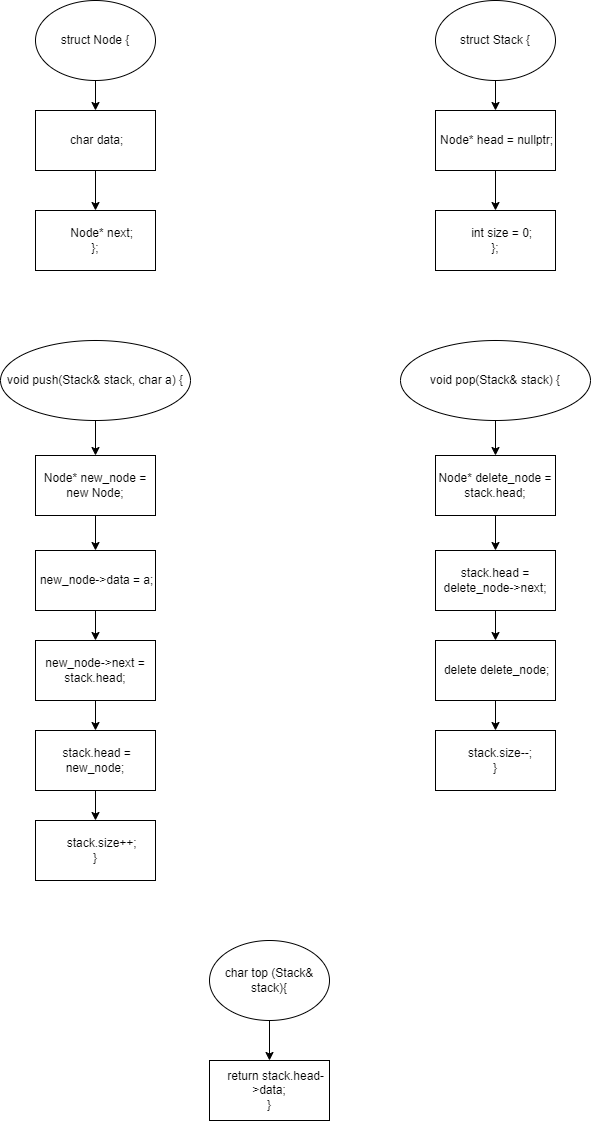
1)Для выполнения данной задачи необходимо создать структуру данных для хранения записей в виде линейного списка. В каждой записи будет содержаться ключевое поле типа \*char (строка символов). Затем необходимо сформировать двунаправленный список, где каждый элемент будет содержать указатель как на предыдущий, так и на следующий элемент.

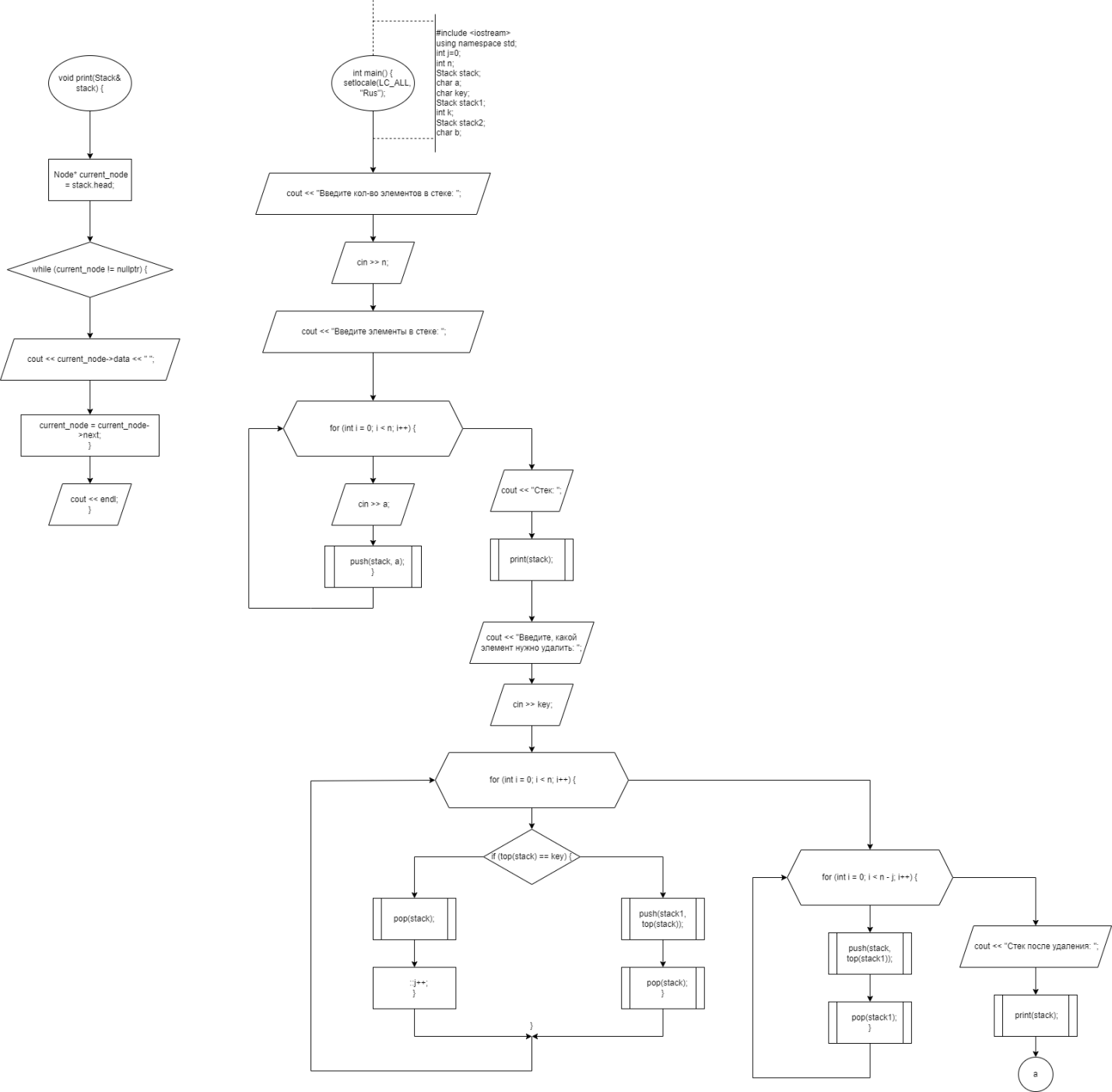
2)После этого нужно реализовать функцию удаления K последних элементов из списка. Для этого можно использовать указатель на конец списка и последовательно освобождать память для K последних элементов.

3)Далее необходимо реализовать функцию добавления элемента после элемента с заданным ключом. Для этого нужно осуществить поиск элемента с заданным ключом в списке, затем создать новый элемент и вставить его после найденного элемента, корректно перенаправив указатели.

4)Таким образом, программа будет создавать и обрабатывать динамические структуры данных в виде двунаправленного списка, удалять элементы из конца списка и добавлять новые элементы после элемента с заданным ключом.

**Блок-Схема:**

****

****

**Код на языке C++:**

#include<iostream>

using namespace std;

int j;

struct Node{

char data;

Node\* next;

};

struct Stack{

Node \* head = nullptr;

int size = 0;

};

void push(Stack& stack, char a) {

Node\* new\_node = new Node;

new\_node->data = a;

new\_node->next = stack.head;

stack.head = new\_node;

stack.size++;

}

void pop(Stack& stack) {

Node\* delete\_node = stack.head;

stack.head = delete\_node->next;

delete delete\_node;

stack.size--;

}

void print(Stack& stack) {

Node\* current\_node = stack.head;

while (current\_node != nullptr) {

cout << current\_node->data << " ";

current\_node = current\_node->next;

}

cout << endl;

}

char top(Stack& stack) {

return stack.head->data;

}

int main() {

setlocale(LC\_ALL, "Rus");

int n;

Stack stack;

char a;

cout << "Введите кол-во элементов в стеке: ";

cin >> n;

cout << "Введите элементы в стеке: ";

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

cin >> a;

push(stack, a);

}

cout << "Стек: ";

print(stack);

char key,key1;

cout << "Введите, значение после которого нужно добавить: ";

cin >> key;

cout << "Введите, значение которое нужно добавить: ";

cin >> key1;

Stack stack1;

bool flag = false;

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

if (flag == true) {

push(stack1, key1);

flag = false;

}

if (top(stack) == key) {

flag = true;

push(stack1, top(stack));

pop(stack);

}

else {

push(stack1, top(stack));

pop(stack);

}

}

int k;

cout << "Введите, сколько элементов надо удалить из стека: ";

cin >> k;

Stack stack2;

n = stack1.size;

for (int i = 0; i < n-k; i++) {

push(stack, top(stack1));

pop(stack1);

}

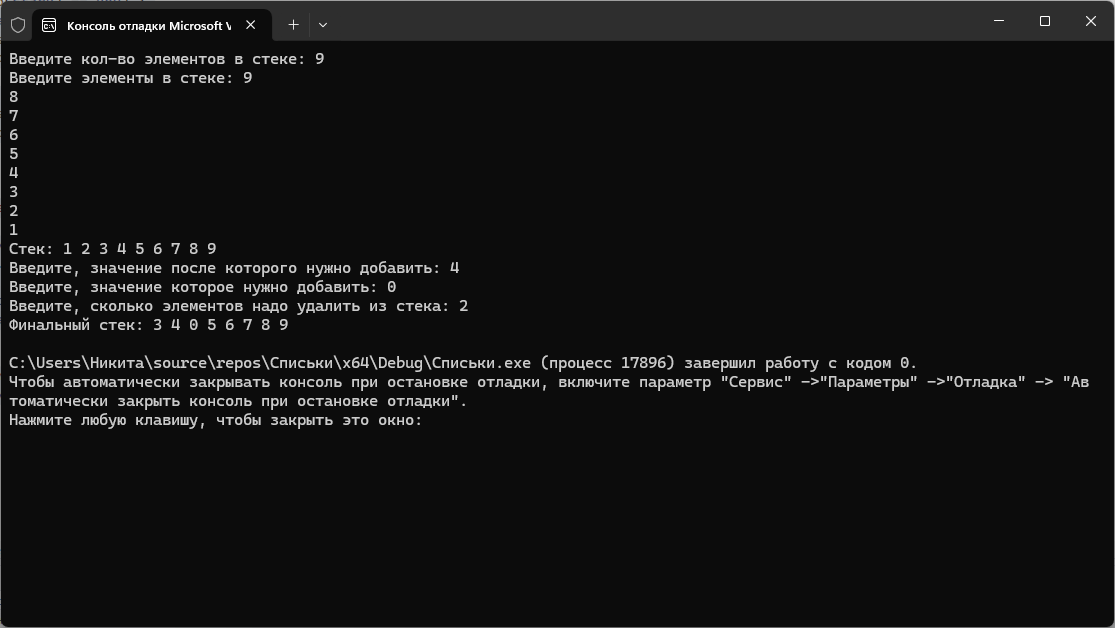
cout << "Финальный стек: ";

print(stack);

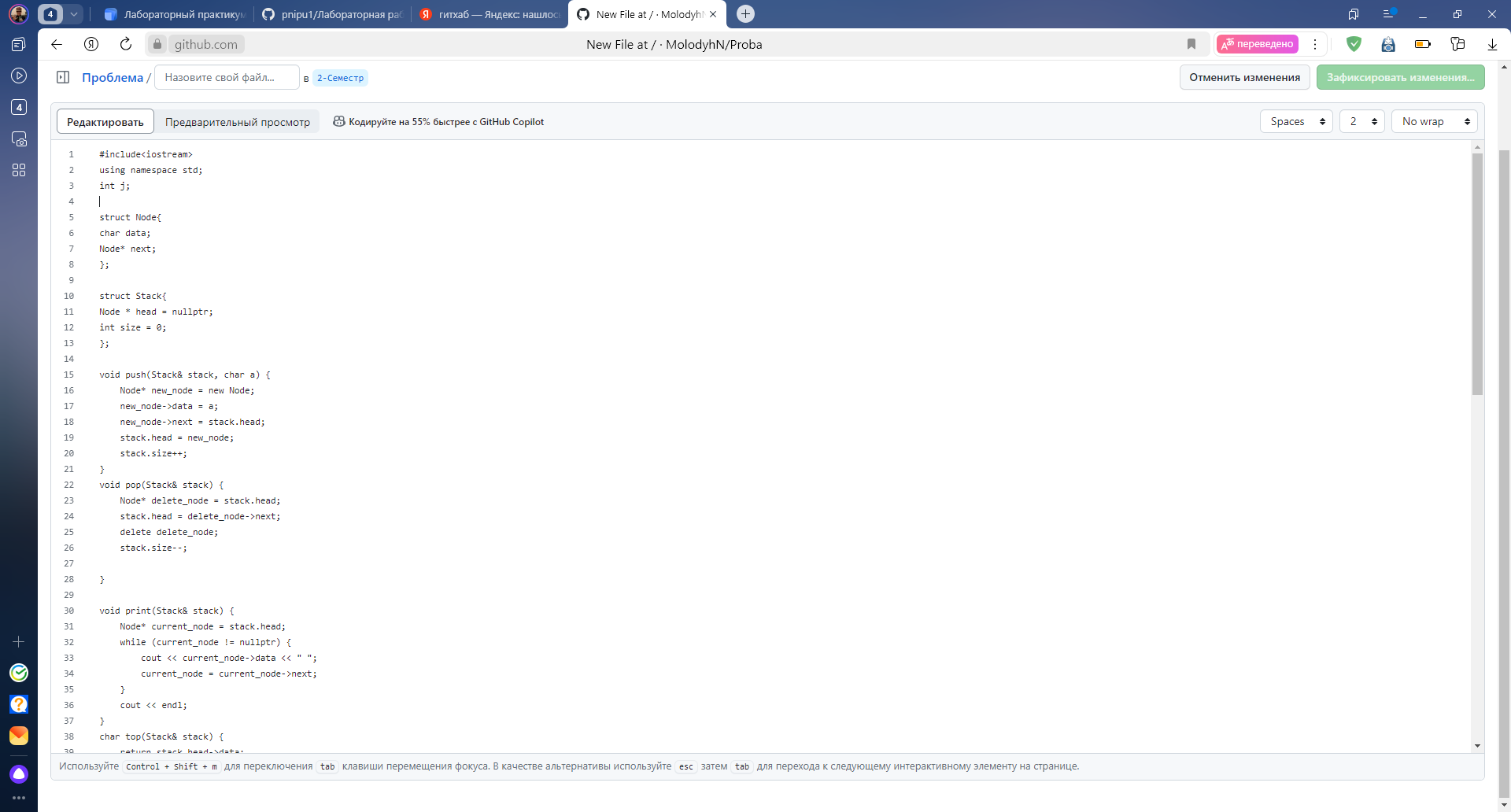
return 0;

}

**Работа программы:**

****

**Cкрины из гита:**

****

https://github.com/MolodyhN/Proba

**Вывод:** Задача была выполнена. Всё получилось.