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:**

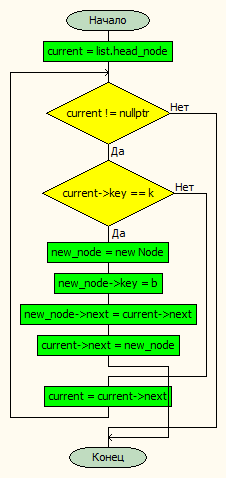
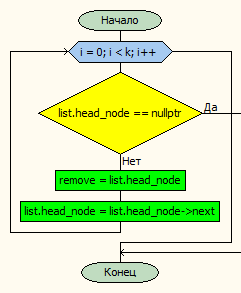
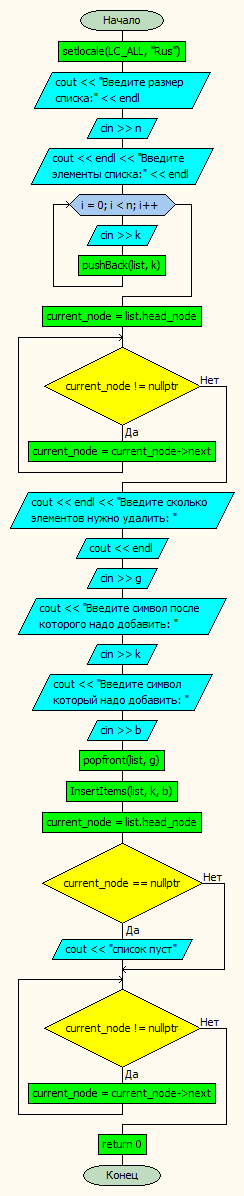
1)Для выполнения данной задачи необходимо создать структуру данных для хранения записей в виде линейного списка. В каждой записи будет содержаться ключевое поле типа \*char (строка символов). Затем необходимо сформировать двунаправленный список, где каждый элемент будет содержать указатель как на предыдущий,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

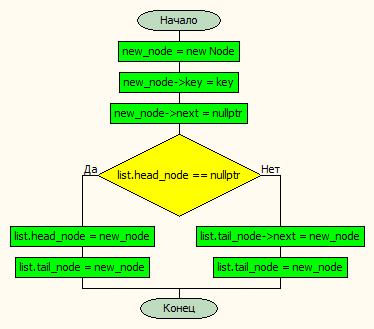
2)После этого нужно реализовать функцию удаления K последних элементов из списка. Для этого можно использовать указатель на конец списка и последовательно освобождать память для K последних элементов.

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;

struct Node {

char key;

Node\* next;

};

struct List {

Node\* head\_node = nullptr;

Node\* tail\_node = nullptr;

};

void pushBack(List& list, char key) {

Node\* new\_node = new Node;

new\_node->key = key;

new\_node->next = nullptr;

if (list.head\_node == nullptr) {

list.head\_node = new\_node;

list.tail\_node = new\_node;

}

else {

list.tail\_node->next = new\_node;

list.tail\_node = new\_node;

}

}

void popfront(List& list, int k) {

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

if (list.head\_node == nullptr) {

return;

}

Node\* remove = list.head\_node;

list.head\_node = list.head\_node->next;

delete remove;

}

}

void InsertItems(List& list, char k, char b) {

Node\* current = list.head\_node;

while (current != nullptr) {

if (current->key == k) {

Node\* new\_node = new Node;

new\_node->key = b;

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

current->next = new\_node;

break; // Added break statement to exit loop after first occurrence

}

current = current->next;

}

}

int main() {

setlocale(LC\_ALL, "Rus");

List list;

cout << "Введите размер списка:" << endl;

int n;

cin >> n;

cout << endl << "Введите элементы списка:" << endl;

char k;

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

cin >> k;

pushBack(list, k);

}

Node\* current\_node = list.head\_node;

while (current\_node != nullptr) {

cout << current\_node->key << ' ';

current\_node = current\_node->next;

}

cout << endl << "Введите сколько элементов нужно удалить: ";

cout << endl;

int g; // Changed type of variable g to int

char b;

cin >> g;

cout << "Введите символ после которого надо добавить: ";

cin >> k; // Changed variable name to k

cout << "Введите символ который надо добавить: ";

cin >> b;

popfront(list, g);

InsertItems(list, k, b);

current\_node = list.head\_node;

if (current\_node == nullptr) { // Changed = to ==

cout << "список пуст";

}

while (current\_node != nullptr) {

cout << current\_node->key << ' ';

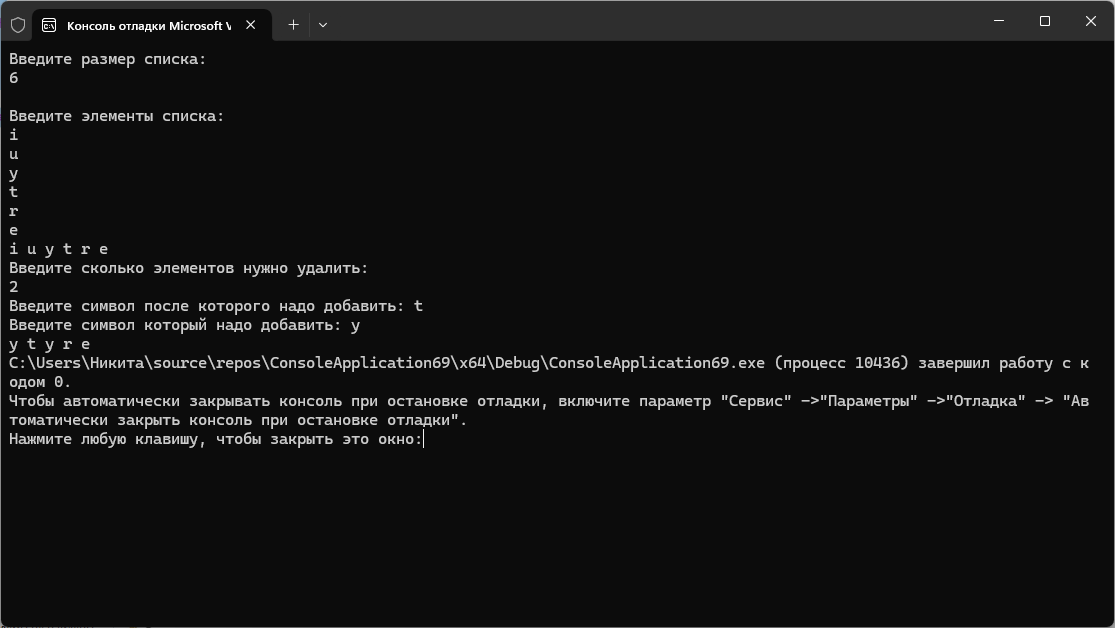
current\_node = current\_node->next;

}

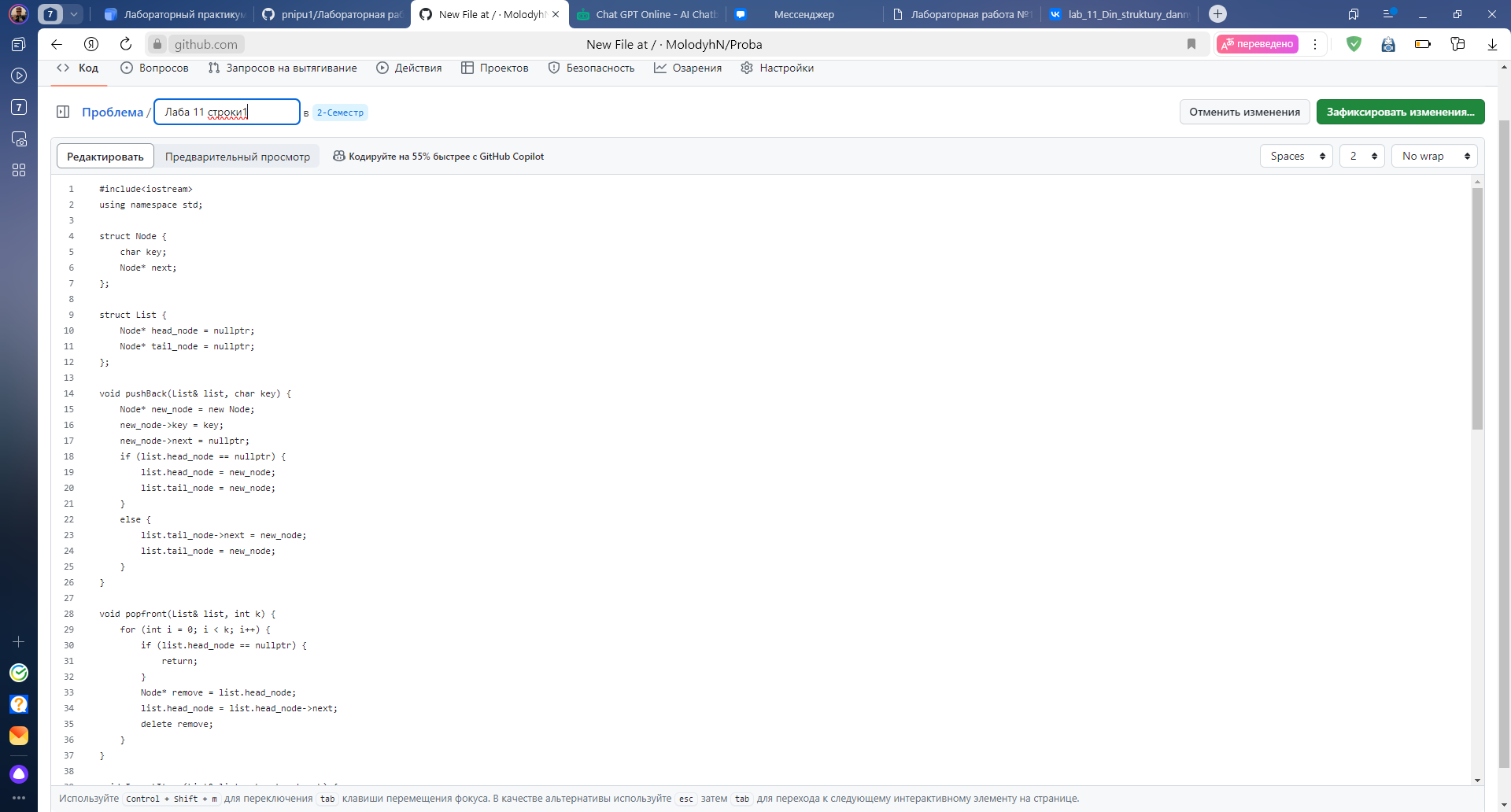
return 0;

}

**Работа программы:**

****

**Cкрины из гита:**

****

https://github.com/MolodyhN/Proba

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