Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

о лабораторной работе

Тема: «Односвязный список»

Выполнил

Студент группы РИС-22-2б

Прядеин И.А.

Проверил доцент кафедры

ИТАС

Полякова О. А.

Пермь 2023

**Постановка задачи**

Вариант 22:

Записи в линейном списке содержат ключевое поле типа \*char(строка символов). Сформировать двунаправленный список. Удалить элемент с заданным ключом. Добавить К элементов перед элементом с заданным ключом.

**Исходный код программы:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**struct Node**

**{**

**string value;**

**Node\* prev, \* next;**

**Node(string val)**

**{**

**value = val;**

**next = nullptr;**

**prev = nullptr;**

**}**

**};**

**struct list**

**{**

**Node\* first, \* last;**

**list()**

**{**

**first = nullptr;**

**last = nullptr;**

**}**

**Node\* push\_front(string val)**

**{**

**Node\* ptr = new Node(val);**

**ptr->next = first;**

**if (first != nullptr)**

**first->prev = ptr;**

**if (last == nullptr)**

**last = ptr;**

**first = ptr;**

**return ptr;**

**}**

**Node\* push\_back(string val)**

**{**

**Node\* ptr = new Node(val);**

**ptr->prev = last;**

**if (last != nullptr)**

**last->next = ptr;**

**if (first == nullptr)**

**first = ptr;**

**last = ptr;**

**return ptr;**

**}**

**void pop\_front()**

**{**

**if (first == nullptr)**

**return;**

**Node\* ptr = first->next;**

**if (ptr != nullptr)**

**ptr->prev = nullptr;**

**else**

**last = ptr;**

**delete first;**

**first = ptr;**

**}**

**void pop\_back()**

**{**

**if (last == nullptr)**

**return;**

**Node\* ptr = last->prev;**

**if (ptr != nullptr)**

**ptr->next = nullptr;**

**else**

**first = nullptr;**

**delete last;**

**last = ptr;**

**}**

**Node\* getEl(string val)**

**{**

**Node\* ptr = first;**

**while (ptr && ptr->value != val)**

**ptr = ptr->next;**

**if (!ptr)**

**cout << "This element does not exist" << endl;**

**return ptr;**

**}**

**void push\_before(string val)**

**{**

**Node\* right = getEl(val);**

**Node\* left = right->prev;**

**int ElNumber;**

**cout << "Enter number of elements: ";**

**cin >> ElNumber;**

**string str;**

**cout << "Enter values: " << endl;**

**cin.ignore();**

**getline(cin, str);**

**Node\* ptr = new Node(str);**

**ptr->prev = left;**

**ptr->next = right;**

**left->next = ptr;**

**right->prev = ptr;**

**left = ptr->prev;**

**right = ptr;**

**for (int i = 1; i < ElNumber; i++)**

**{**

**getline(cin, str);**

**Node\* ptr = new Node(str);**

**ptr->prev = left;**

**ptr->next = right;**

**left->next = ptr;**

**right->prev = ptr;**

**left = ptr->prev;**

**right = ptr;**

**}**

**}**

**Node\* insert(string val, string data)**

**{**

**Node\* right = getEl(val);**

**if (right == nullptr)**

**return push\_back(data);**

**Node\* left = right->prev;**

**if (left == nullptr)**

**return push\_front(data);**

**Node\* ptr = new Node(data);**

**ptr->prev = left;**

**ptr->next = right;**

**left->next = ptr;**

**right->prev = ptr;**

**return ptr;**

**}**

**void popEl(string val)**

**{**

**Node\* ptr = getEl(val);**

**if (ptr == nullptr)**

**return;**

**if (ptr->prev == nullptr)**

**{**

**pop\_front();**

**return;**

**}**

**if (ptr->next == nullptr)**

**{**

**pop\_back();**

**return;**

**}**

**Node\* left = ptr->prev;**

**Node\* right = ptr->next;**

**left->next = right;**

**right->prev = left;**

**delete ptr;**

**}**

**void deleteAll()**

**{**

**while (first != nullptr)**

**pop\_front();**

**}**

**void print()**

**{**

**if (first == nullptr)**

**{**

**cout << "List is empty" << endl;**

**return;**

**}**

**Node\* ptr = first;**

**cout << "List data: ";**

**while (ptr)**

**{**

**cout << ptr->value << " ";**

**ptr = ptr->next;**

**}**

**cout << endl;**

**}**

**};**

**void showMenu()**

**{**

**cout << "\tMenu\n";**

**cout << "Enter 0 to end the program\n";**

**cout << "Enter 1 to view the list\n";**

**cout << "Enter 2 to create a list\n";**

**cout << "Enter 3 to add the item to the top of the list\n";**

**cout << "Enter 4 to add an item to the end of the list\n";**

**cout << "Enter 5 to add multiple elements before the element with the given number\n";**

**cout << "Enter 6 to remove an element with given value\n";**

**cout << "Enter 7 to delete the list\n";**

**}**

**void menu()**

**{**

**int switch\_n, size\_list;**

**string str;**

**list l;**

**do {**

**system("pause");**

**system("cls");**

**showMenu();**

**cin >> switch\_n;**

**switch(switch\_n)**

**{**

**case 0:**

**cout << "The program has ended";**

**break;**

**case 1:**

**l.print();**

**break;**

**case 2:**

**cout << "Enter list size: ";**

**cin >> size\_list;**

**cin.ignore();**

**for (int i = 0; i < size\_list; i++)**

**{**

**getline(cin, str);**

**l.push\_back(str);**

**}**

**break;**

**case 3:**

**cout << "Enter value: ";**

**cin.ignore();**

**getline(cin, str);**

**l.push\_front(str);**

**break;**

**case 4:**

**cout << "Enter value: ";**

**cin.ignore();**

**getline(cin, str);**

**l.push\_back(str);**

**break;**

**case 5:**

**cout << "Enter value: ";**

**cin.ignore();**

**getline(cin, str);**

**l.push\_before(str);**

**break;**

**case 6:**

**cout << "Enter value: ";**

**cin.ignore();**

**getline(cin, str);**

**l.popEl(str);**

**break;**

**case 7:**

**l.deleteAll();**

**break;**

**}**

**}**

**while (switch\_n != 0);**

**}**

**int main()**

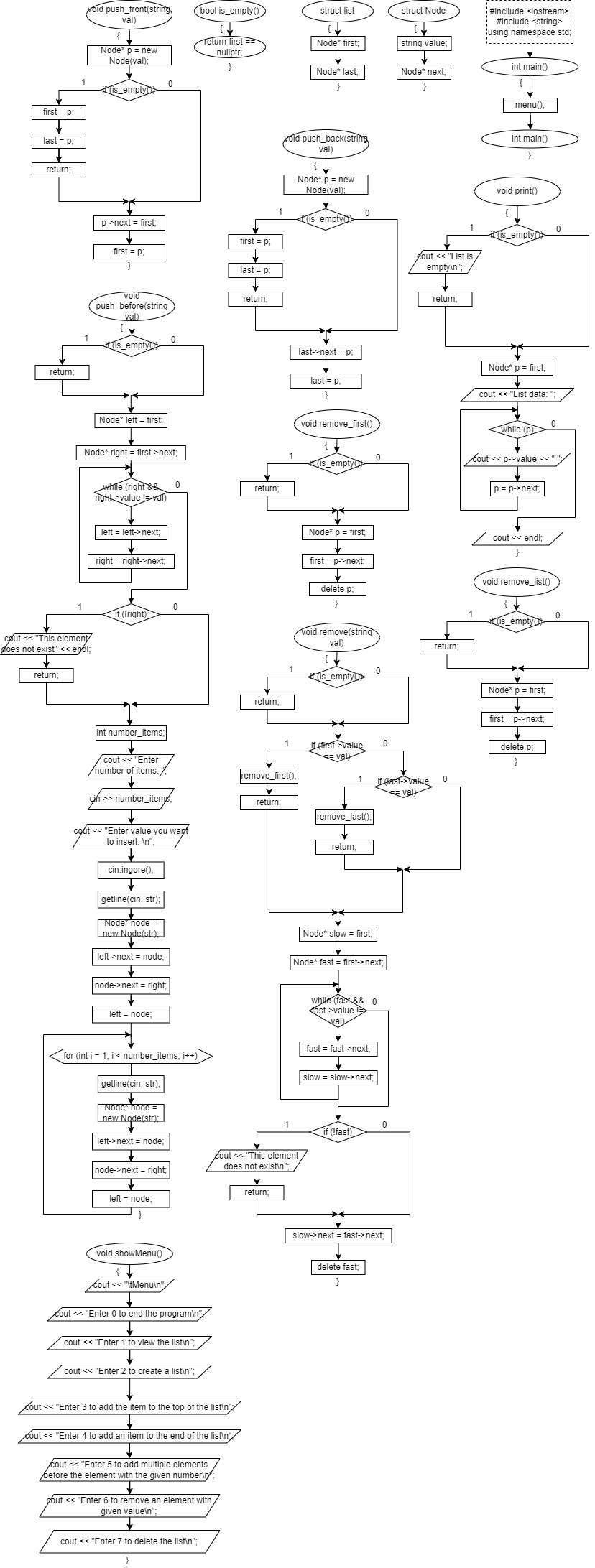
**{**

**menu();**

**return 0;**

**}**

**Блок схема**



**Скриншоты результатов выполнения программы**

