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\* next;

Node(string val)

{

value = val;

next = nullptr;

}

};

struct list

{

Node\* first;

Node\* last;

list()

{

first = nullptr;

last = nullptr;

}

bool is\_empty()

{

return first == nullptr;

}

void push\_front(string val)

{

Node\* p = new Node(val);

if (is\_empty())

{

first = p;

last = p;

return;

}

p->next = first;

first = p;

}

void push\_back(string val)

{

Node\* p = new Node(val);

if (is\_empty())

{

first = p;

last = p;

return;

}

last->next = p;

last = p;

}

void push\_before(string val)

{

if (is\_empty())

return;

Node\* left = first;

Node\* right = first->next;

while (right && right->value != val)

{

left = left->next;

right = right->next;

}

if (!right)

{

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

return;

}

int number\_items;

cout << "Enter number of items: ";

cin >> number\_items;

string str;

cout << "Enter value you want to insert: \n";

cin.ignore();

getline(cin, str);

Node\* node = new Node(str);

left->next = node;

node->next = right;

left = node;

for (int i = 1; i < number\_items; i++)

{

getline(cin, str);

node = new Node(str);

left->next = node;

node->next = right;

left = node;

}

}

void print()

{

if (is\_empty())

{

cout << "List is empty\n";

return;

}

Node\* p = first;

cout << "List data: ";

while (p)

{

cout << p->value << " ";

p = p->next;

}

cout << endl;

}

void remove\_first()

{

if (is\_empty())

return;

Node\* p = first;

first = p->next;

delete p;

}

void remove\_last()

{

if (is\_empty())

return;

if (first == last)

{

remove\_first();

return;

}

Node\* p = first;

while (p->next != last)

p = p->next;

p->next = nullptr;

delete last;

last = p;

}

void remove(string val)

{

if (is\_empty())

return;

if (first->value == val)

{

remove\_first();

return;

}

else if (last->value == val)

{

remove\_last();

return;

}

Node\* slow = first;

Node\* fast = first->next;

while (fast && fast->value != val)

{

fast = fast->next;

slow = slow->next;

}

if (!fast)

{

cout << "This element does not exist\n";

return;

}

slow->next = fast->next;

delete fast;

}

void remove\_list()

{

if (is\_empty())

return;

Node\* p = first;

while (p)

{

remove\_first();

p = p->next;

}

}

};

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.remove(str);

break;

case 7:

l.remove\_list();

break;

}

}

while (switch\_n != 0);

}

int main()

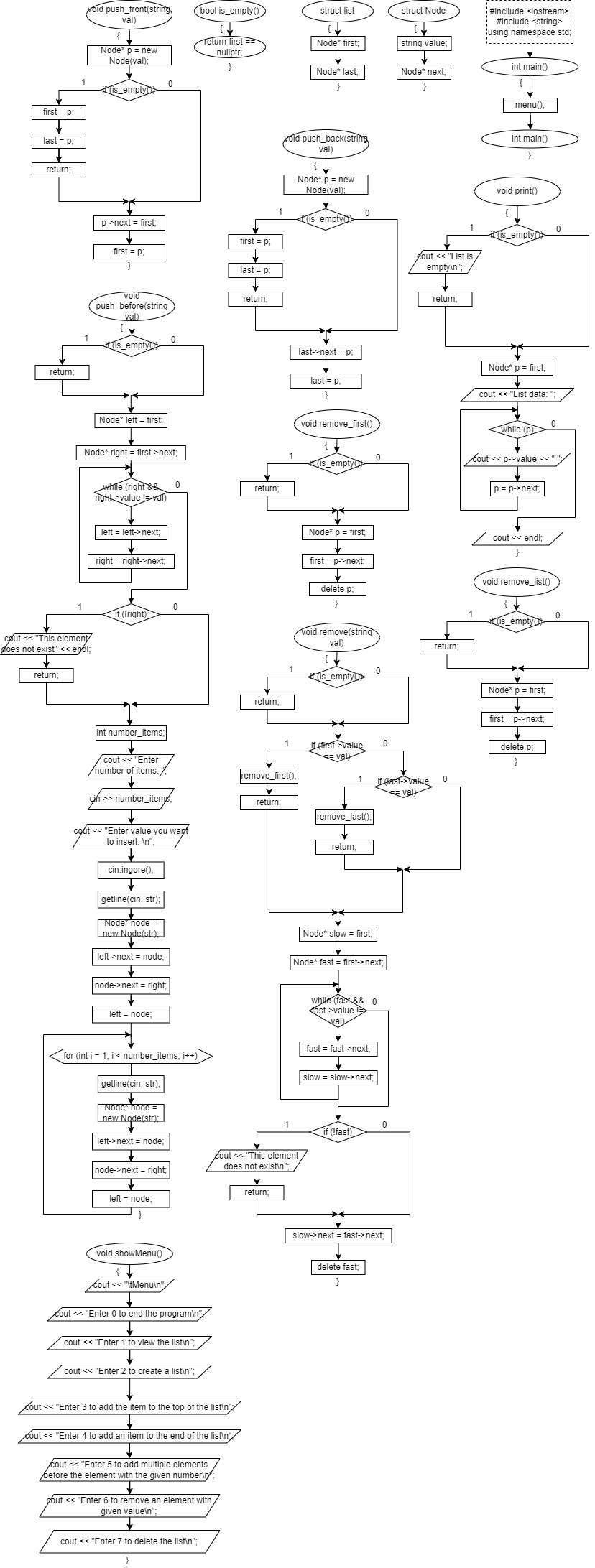
{

menu();

return 0;

}

**Блок схема**



**Скриншоты результатов выполнения программы**

