Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**О Т Ч Е Т**

**по лабораторной работе №11.3 (Стеки)**

**по дисциплине**

**«Основы алгоритмизации и программирования» семестр 2**

Выполнил студент гр. ИВТ-21-1б

Ипатов Дмитрий Сергеевич

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Д.В.

(оценка) (подпись)

г. Пермь-2022

**Постановка задачи:**

1. Сформировать стек. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

Выполнить двумя способами (без библиотеки stack и с ней).

Анализ задачи:

1 способ

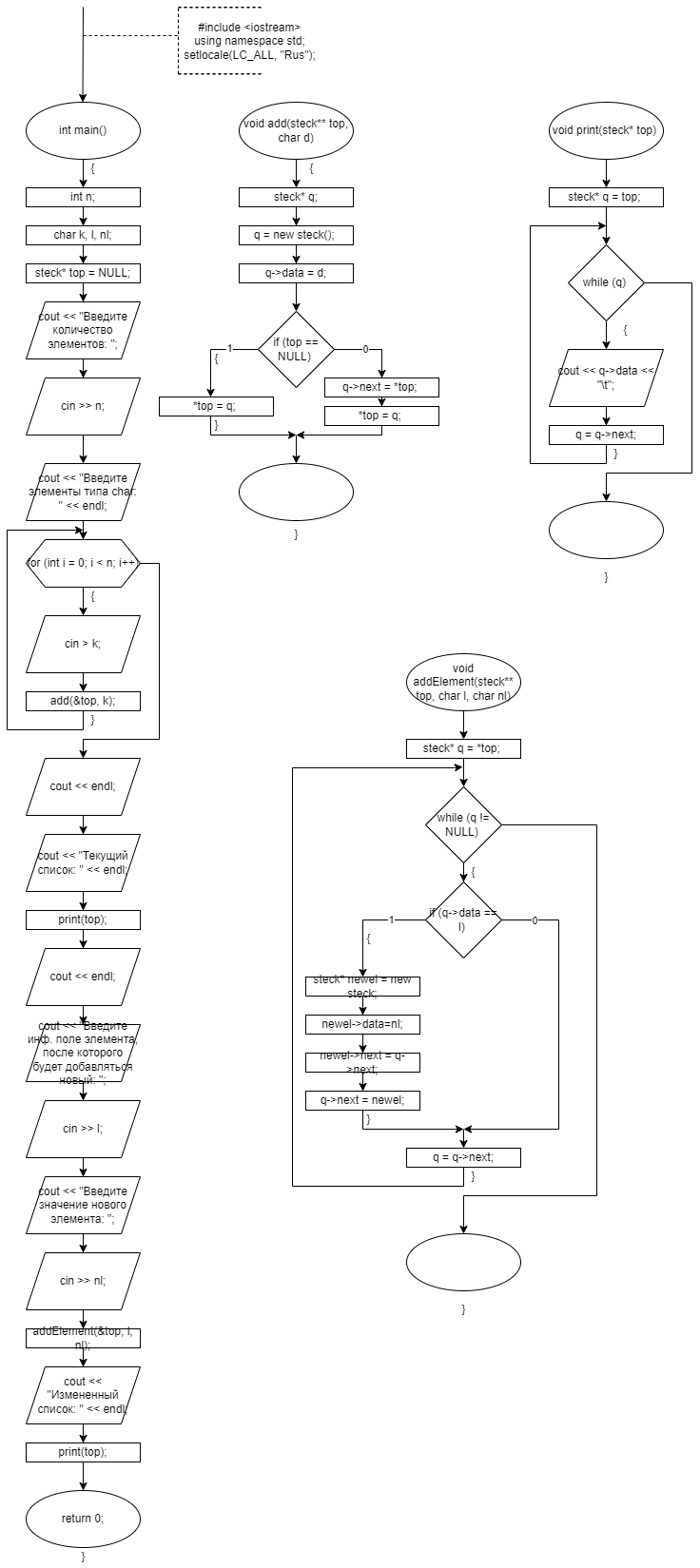
1. Создаем структуру stack с информационным полем char и с указателем на следующий элемент;
2. В функции main просим пользователя ввести количество элементов, сами эти элементы через цикл, которые будут отправляться в функцию add (в которой элементы будут создаваться в структуре и иметь указатели на следующий элемент в структуре);
3. Выводим текущий стэк с помощью функции print (в которой пока q не ссылается на нулевой указатель будет происходить вывод информационного поля элемента и переход на следующий элемент);
4. Дальше просим пользователя определить после какого значения будет добавляться новый элемент и значение этого элемента;
5. Далее вызываем функцию addElement, в которую передаем начало стэка, элемент после которого добавить и значение нового элемента;
6. После того как программа добавила элемент(ы) выводим новый (измененный) стэк.

2 способ

1. Подключаем библиотеку stack, в функции main создаем стэк с типом данных char, под названием st;
2. Просим пользователя ввести количество элементов, после элемента с каким значением добавить и элемент с каким значением добавить;
3. На этапе ввода проверяем совпадает ли введенный элемент с тем, после которого нужно добавить, если совпадает, то сначала пушим элемент, который нужно добавить, а потом тот, который только что ввели, если же не совпадает, то пушим только введенный элемент;
4. Выводим полученный стэк.

**1 способ (без библиотеки stack)**

**Блок схема:**

****

**Код программы:**

#include <iostream>

using namespace std;

struct steck

{

char data;

steck\* next;

};

void add(steck\*\* top, char d)

{

steck\* q;

q = new steck();

q->data = d;

if (top == NULL)

{

\*top = q;

}

else

{

q->next = \*top;

\*top = q;

}

}

void print(steck\* top)

{

steck\* q = top;

while (q)

{

cout << q->data << "\t";

q = q->next;

}

}

void addElement(steck\*\* top, char l, char nl)

{

steck\* q = \*top;

while (q != NULL)

{

if (q->data == l)

{

steck\* newel = new steck;

newel->data=nl;

newel->next = q->next;

q->next = newel;

}

q = q->next;

}

}

int main()

{

int n;

char k, l, nl;

setlocale(LC\_ALL, "Rus");

steck\* top = NULL;

cout << "Введите количество элементов: ";

cin >> n;

cout << "Введите элементы типа char: " << endl;

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

{

cin >> k;

add(&top, k);

}

cout << endl;

cout << "Текущий список: " << endl;

print(top);

cout << endl;

cout << "Введите инф. поле элемента, после которого будет добавляться новый: ";

cin >> l;

cout << "Введите значение нового элемента: ";

cin >> nl;

addElement(&top, l, nl);

cout << "Измененный список: " << endl;

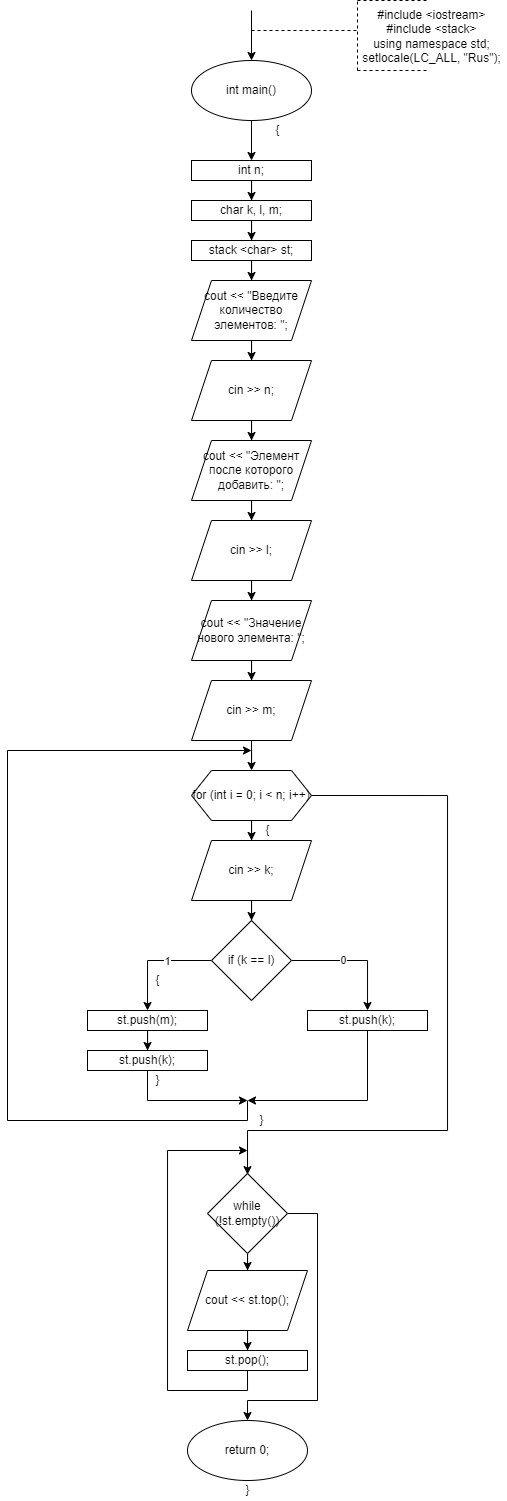
print(top);

return 0;

}

**2 способ (с библиотекой stack)**

**Блок схема:**

****

**Код программы:**

#include <iostream>

#include <stack>

using namespace std;

int main()

{

int n;

char k, l, m;

setlocale(LC\_ALL, "Rus");

stack <char> st;

cout << "Введите количество элементов: ";

cin >> n;

cout << "Элемент после которого добавить: ";

cin >> l;

cout << "Значение нового элемента: ";

cin >> m;

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

{

cin >> k;

if (k == l)

{

st.push(m);

st.push(k);

}

else

{

st.push(k);

}

}

while (!st.empty())

{

cout << st.top();

st.pop();

}

return 0;

}