Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное 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»

**Лабораторная работа № 11**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**на тему**

**«Динамические структуры данных»**

**Вариант 25**

Выполнил студент гр. ИВТ-20-1б

Камянецький Сергей Александрович\_\_

Проверил:

Доцент кафедры ИТАС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Полякова О. А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пермь 2020

**Содержание**

[Цель и задачи работы 3](#_Toc64562995)

[Реализация через STL-библиотеки 4](#_Toc64562996)

[Анализ 4](#_Toc64562997)

[Блок-схемы 6](#_Toc64562998)

[Код 10](#_Toc64562999)

[Результат работы 14](#_Toc64563000)

[Реализация через структуры 15](#_Toc64563001)

[Анализ 15](#_Toc64563002)

[Блок-схемы 17](#_Toc64563003)

[Код 24](#_Toc64563004)

[Результат работы 30](#_Toc64563005)

# Цель и задачи работы

Цель работы:

Знакомство с динамическими информационными структурами на примере стека.

Общие задачи работы:

Написать программу, в которой создаются динамические структуры и выполнить их обработку в соответствии со своим вариантом.

Задачи 25 варианта работы:

Записи в стеке содержат ключевое поле типа char(строка символов). Сформировать стек.

Удалить элемент с заданным ключом.

Добавить К элементов перед элементом с заданным номером.

# Реализация через STL-библиотеки

## Анализ

Так как нужно реализовать программу через STL – библиотеку, нужно подключить библиотеку <stack> .

Из поставленной задачи можно сделать вывод, что нужно выводить стек несколько раз, поэтому, чтобы не писать несколько раз одно и тоже, создадим функцию (void print(stack<char> st)) для вывода элементов стека. Введём переменную (int p), которая будет считать размер стека с помощью (size). После чего с помощью оператора выбора (if) проверим пуст ли стек или нет. Если да, то на экран должно вывестись сообщение, что стек пуст, иначе с помощью цикла(for (int i = 0; i < p; i++)) должны выводиться элементы стека. Чтобы выводились все элементы, надо после вывода вершины стека удалить элемент из стека.

Записи в стеке содержат ключевое поле типа char(строка символов). Сформировать стек:

Так как нужно записывать элементы в стек, надо его создать (stack<char> st;). После чего, так как не сказано сколько элементов содержит стек, пользователь должен ввести (int n-количество элементов в стеке). Затем с помощью цикла(for (int i = 0; i < n; i++)) введем элемент стека в переменную (char a) и добавим его в стек. После чего с помощью ранее написанной функции (print) выведем элементы стека на экран.

Так как нужно реализовать программу через STL – библиотеку, нужно подключить библиотеку <stack> .

Из поставленной задачи можно сделать вывод, что нужно выводить стек несколько раз, поэтому, чтобы не писать несколько раз одно и тоже, создадим функцию (void print(stack<char> st)) для вывода элементов стека. Введём переменную (int p), которая будет считать размер стека с помощью (size). После чего с помощью оператора выбора (if) проверим пуст ли стек или нет. Если да, то на экран должно вывестись сообщение, что стек пуст, иначе с помощью цикла(for (int i = 0; i < p; i++)) должны выводиться элементы стека. Чтобы выводились все элементы, надо после вывода вершины стека удалить элемент из стека.

Записи в стеке содержат ключевое поле типа char(строка символов). Сформировать стек:

Так как нужно записывать элементы в стек, надо его создать (stack<char> st;). После чего, так как не сказано сколько элементов содержит стек, пользователь должен ввести (int n-количество элементов в стеке). Затем с помощью цикла(for (int i = 0; i < n; i++)) введем элемент стека в переменную (char a) и добавим его в стек. После чего с помощью ранее написанной функции (print) выведем элементы стека на экран.

## Блок-схемы

# 

# 

# 

# 

## Код

#include <iostream>

#include <stack>

using namespace std;

char a,b;

void print(stack<char> st) {

int p = st.size();

if (p == 0)

{

cout << "стек пуст";

}

else {

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

{

cout << st.top() << " ";

st.pop();

}

}

cout << endl;

}

int main() {

setlocale(LC\_ALL, "Rus");

stack<char> st;

int n;

cout << "Введите количество элементов стека : ";

cin >> n;

cout << "Введите элементы стека : ";

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

{

cin >> a;

st.push(a);

}

cout << "Стек: ";

print(st);

stack<char> st2;

cout << "Введите ключ для удаления: ";

cin >> b;

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

{

if (st.top() != b)

{

st2.push(st.top());

}

st.pop();

}

n = st2.size();

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

{

st.push(st2.top());

st2.pop();

}

cout << "Стек после удаления: ";

print(st);

int k = 0;

cout << "Введите количество элементов для добавления: ";

cin >> k;

int s = 0;

cout << "Введите номер элемента, перед которым добавляют элементы: ";

cin >> s;

for (int i = 0; i < n-s+1; i++)

{

st2.push(st.top());

st.pop();

}

cout << "Введите элементы: ";

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

{

cin >> a;

st.push(a);

}

for (int i = 0; i < n - s + 1; i++)

{

st.push(st2.top());

st2.pop();

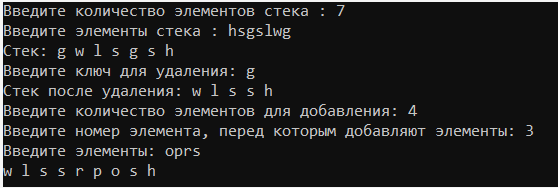
}

print(st);

return 0;

}

## Результат работы



# Реализация через структуры

## Анализ

Так как нужно реализовать программу через структуры, нужно создать структуру, которая будет содержать поля (char data;) для данных и (Stack\* prev;) для адреса следующего элемента.

Создам функцию (Stack\* make\_stack(int n)) для создания стека и его заполнения. В начале проверим с помощью оператора выбора (if) равно ли количество элементов стека 0. Если да, то возвращаем NULL. Иначе выделим память под 1 элемент (p = new Stack;), пользователь записывает данные (cin >> a; p->data = (a);)и адрес на NULL (p->prev = NULL; ) в элемент. Затем ставим указатель на верхний элемент стека (top = p;).

После этого с помощью цикла (for (int i = 2; i <= n; i++)) введём оставшиеся элементы. (пользователь записывает данные и адрес на предыдущий элемент в элемент) В конце возвращаем указатель на верхний элемент.

Создам функцию (void print\_stack(Stack\* top)) для вывода элементов стека. В начале проверим с помощью оператора выбора (if) указывает ли верхний элемент стека (top) на NULL. Если да, то выводим на экран сообщение “стек пуст”. Иначе с помощью (while) выводим элемент и переходим к следующему пока указатель не будет равен NULL.

Создам функцию (char pop(Stack\*& top)) для возвращения первого элемента и его удаления. Для начала считаем количество элементов в стеке (k) с помощью (while). Потом с помощью (if) проверяем равно ли (k) единице. Если да, то обнуляем указатель и возвращаем элемент. Иначе сохраняем значение последнего элемента, делаем второй элемент первым, удаляем последний элемент и возвращаем первый элемент.

Создам функцию (Stack\* push(Stack\*& top, char val)) для добавления элемента в стек. Сначала выделим память под новый элемент. Потом присваиваем значение , которое вводит пользователь, для нового элемента, делаем указатель на нижний элемент и делаем новый элемент первым элементом стека.

Записи в стеке содержат ключевое поле типа char(строка символов). Сформировать стек:

Так как не сказано сколько элементов содержит стек, то пользователь должен ввести количество (n). Потом вызвать ранее написанные функцию для создания стека и его заполнения (Stack\* st = make\_stack(n);) и функцию для вывода стека (print\_stack(st);).

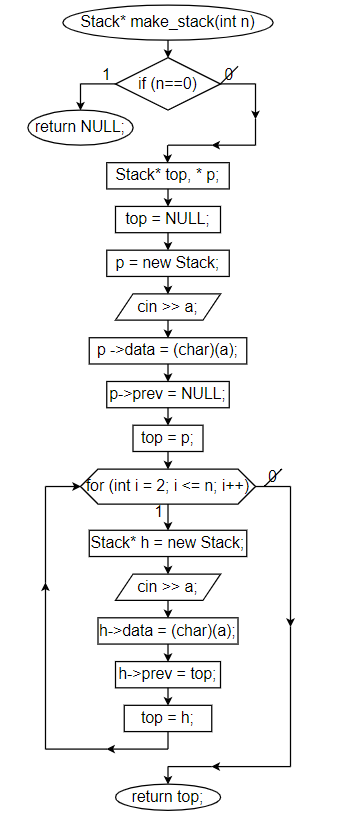
Удалить элемент с заданным ключом:

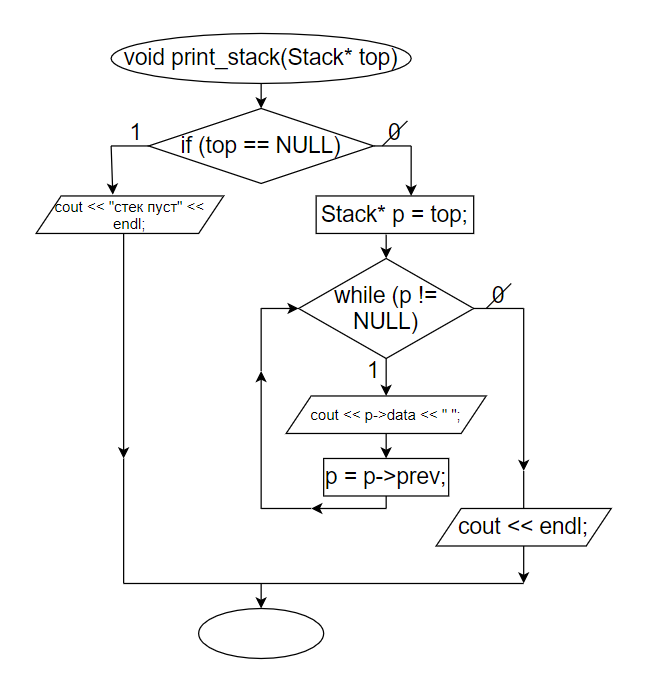
Чтобы выполнить поставленную задачу создадим второй стек, в который будут переноситься нужные элементы главного стека (Stack\* st2 = make\_stack(0);). Потом, так как ключ не указан, пользователь должен ввести ключ (b) для удаления. После чего с помощью цикла (for (int i = 0; i < n; i++)) и оператора выбора (if) перенесём все нужные элементы во второй стек и посчитаем количество элементов равных ключу (k). Для того, чтобы определить количество элементов после удаления вычтим из n-k. Затем с помощью цикла (for (int i = 0; i < n; i++) ) перенесём элементы из второго стека в исходный. После чего с помощью ранее написанной функции (print) выведем элементы стека на экран.

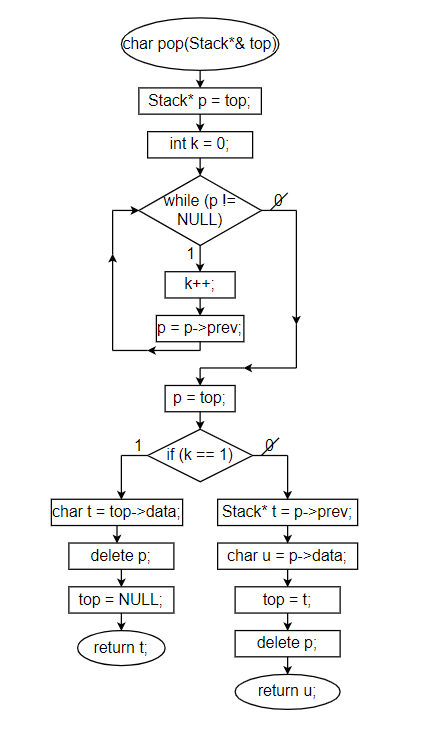
Добавить К элементов перед элементом с заданным номером:

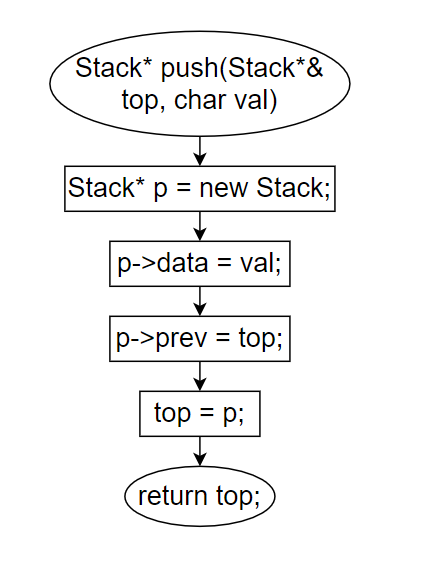
Так как количество элементов для добавления (k) и номер элемента (s), перед которым добавляют элементы,, не указаны, пользователь должен ввести (k) и (s). После этого перенесём с помощью цикла (for (int i = 0; i < n-s+1; i++)) (n-s+1, так как нужно перенести все элементы включая (s) в стек) нужные элементы во второй стек. Затем с помощью цикла (for (int i = 0; i < k; i++)) добавим (k) элементов в исходный стек. После чего с помощью цикла (for (int i = 0; i < n - s + 1; i++)) перенесём элементы из второго стека в исходный и с помощью ранее написанной функции (print) выведем элементы стека на экран.

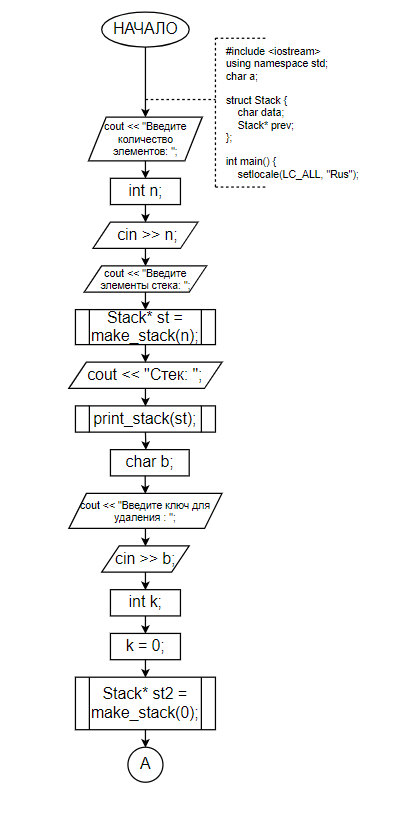
## Блок-схемы

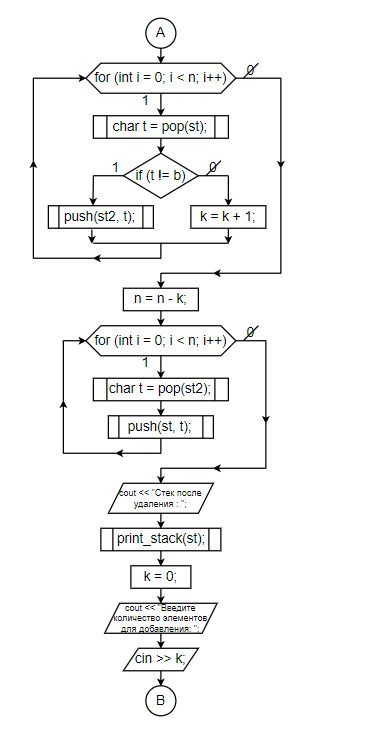


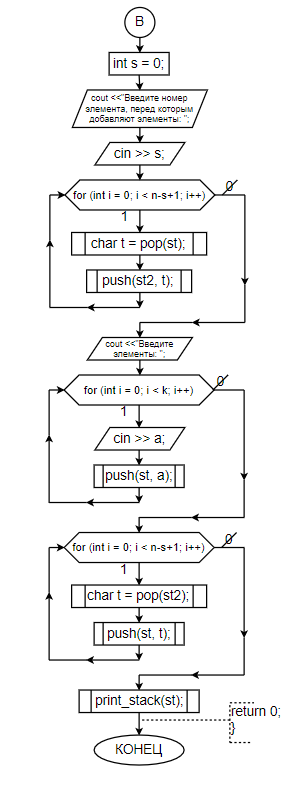












## Код

#include <iostream>

using namespace std;

char a;

struct Stack {

char data;

Stack\* prev;

};

Stack\* make\_stack(int n)

{//создание стека

if (n == 0) return NULL;

Stack\* top, \* p;

top = NULL;

p = new Stack;

cin >> a;

p->data = a;

p->prev = NULL;

top = p;

for (int i = 2; i <= n; i++) {

Stack\* h = new Stack;

cin >> a;

h->data = a;

h->prev = top;

top = h;

}

return top;

}

void print\_stack(Stack\* top)

{//печать стека

if (top == NULL)

{

cout << "стек пуст" << endl;

}

else {

Stack\* p = top;

while (p != NULL)

{

cout << p->data << "

p = p->prev;

}

cout << endl;

}

}

char pop(Stack\*& top)

{

Stack\* p = top;

int k = 0;

while (p != NULL)

{

k++;

p = p->prev;

}

p = top;

if (k == 1) {

char t = top->data;

delete p;

top = NULL;

return t;

}

else

{

Stack\* t = p->prev;

char u = p->data;

top = t;

delete p;

return u;

}

}

Stack\* push(Stack\*& top, char val)

{

Stack\* p = new Stack;

p->data = val;

p->prev = top;

top = p;

return top;

}

int main() {

setlocale(LC\_ALL, "Rus");

cout << "Введите количество элементов: ";

int n;

cin >> n;

cout << "Введите элементы стека: ";

Stack\* st = make\_stack(n);

cout << "Стек: ";

print\_stack(st);

char b;

cout << "Введите ключ для удаления : ";

cin >> b;

int k;

k = 0;

Stack\* st2 = make\_stack(0);

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

{

char t = pop(st);

if (t != b)

{

push(st2, t);

}

else

{

k = k + 1;

}

}

n = n - k;

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

{

char t = pop(st2);

push(st, t);

}

cout << "Стек после удаления : ";

print\_stack(st);

k = 0;

cout << "Введите количество элементов для добавления: ";

cin >> k;

int s = 0;

cout << "Введите номер элемента, перед которым добавляют элементы: ";

cin >> s;

for (int i = 0; i < n - s + 1; i++)

{

char t = pop(st);

push(st2, t);

}

cout << "Введите элементы: ";

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

{

cin >> a;

push(st, a);

}

for (int i = 0; i < n - s + 1; i++)

{

char t = pop(st2);

push(st, t);

}

print\_stack(st);

return 0;

}

## Результат работы

