Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема : "Стеки"

Выполнила работу:

Студентка группы РИС-22-1Б

Верхоланцева Е.С.

Проверила:

Доцент кафедры ИТАС

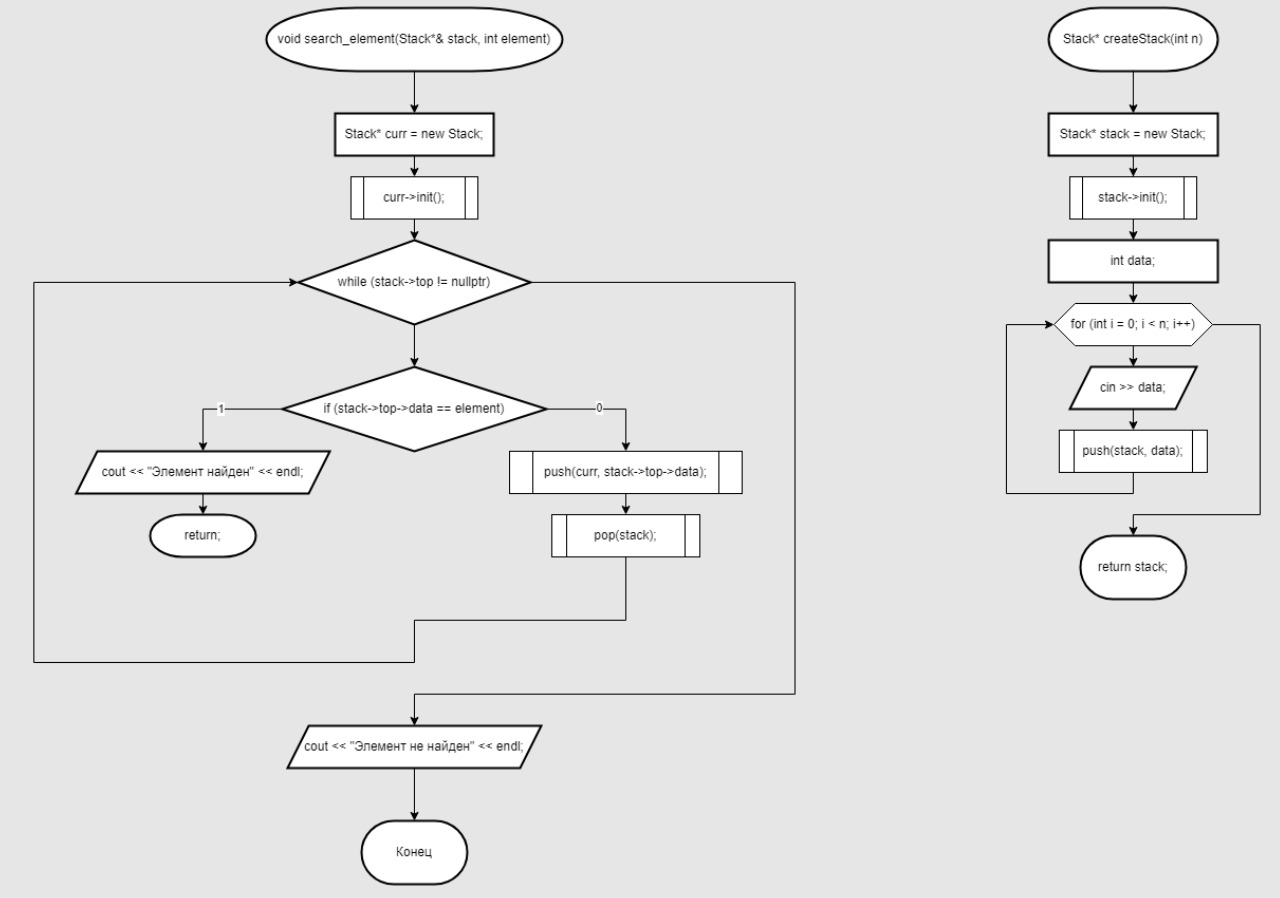
Полякова О.А.

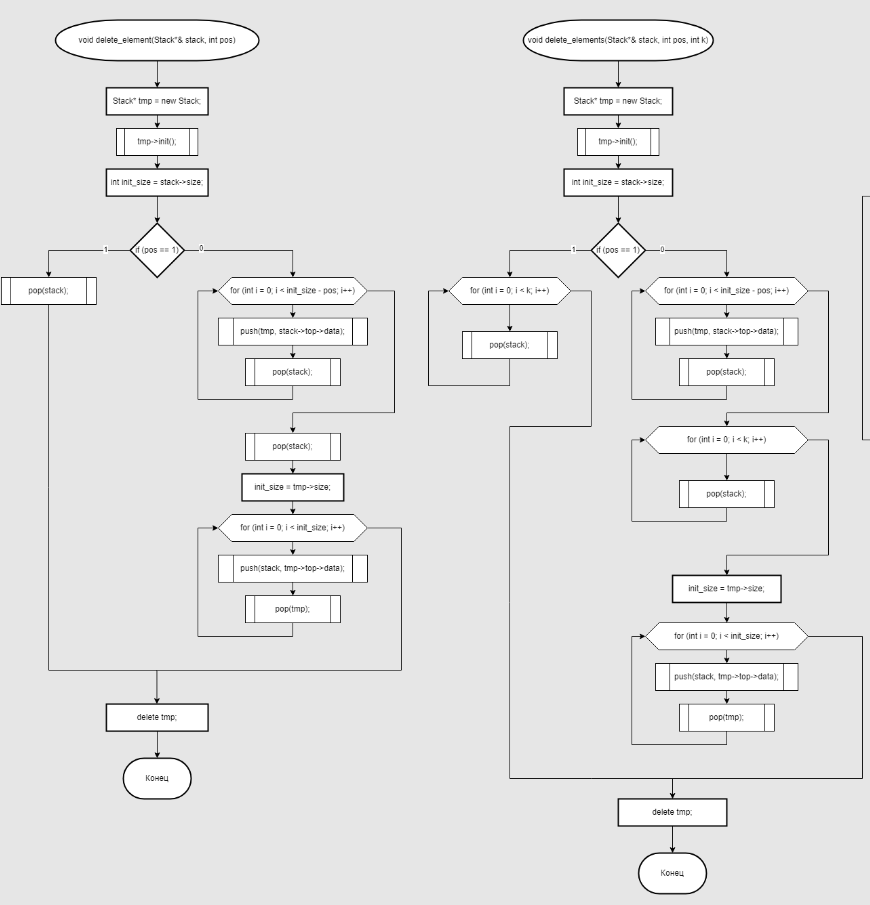
Г. Пермь-2023

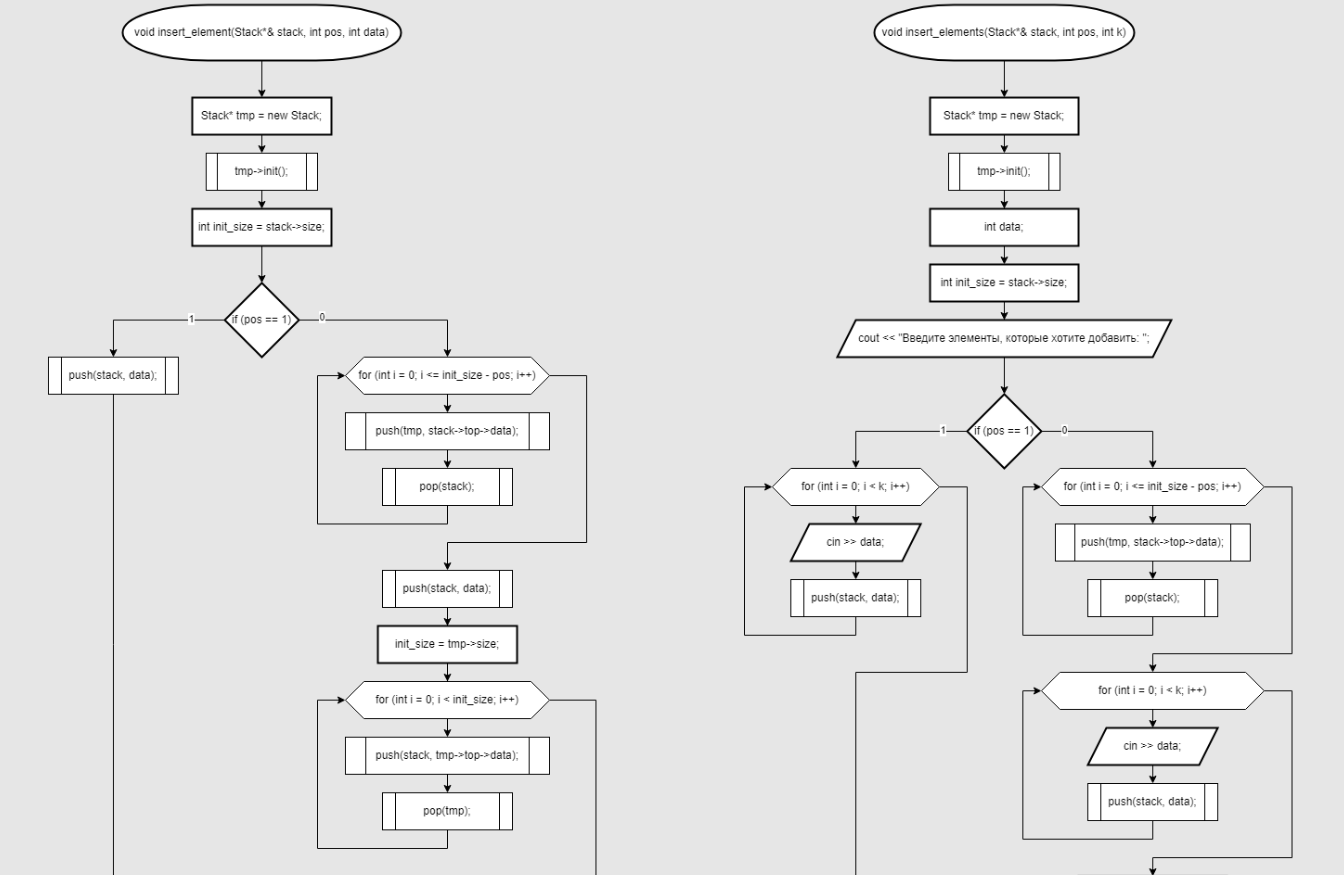
**1. Постановка задачи.**

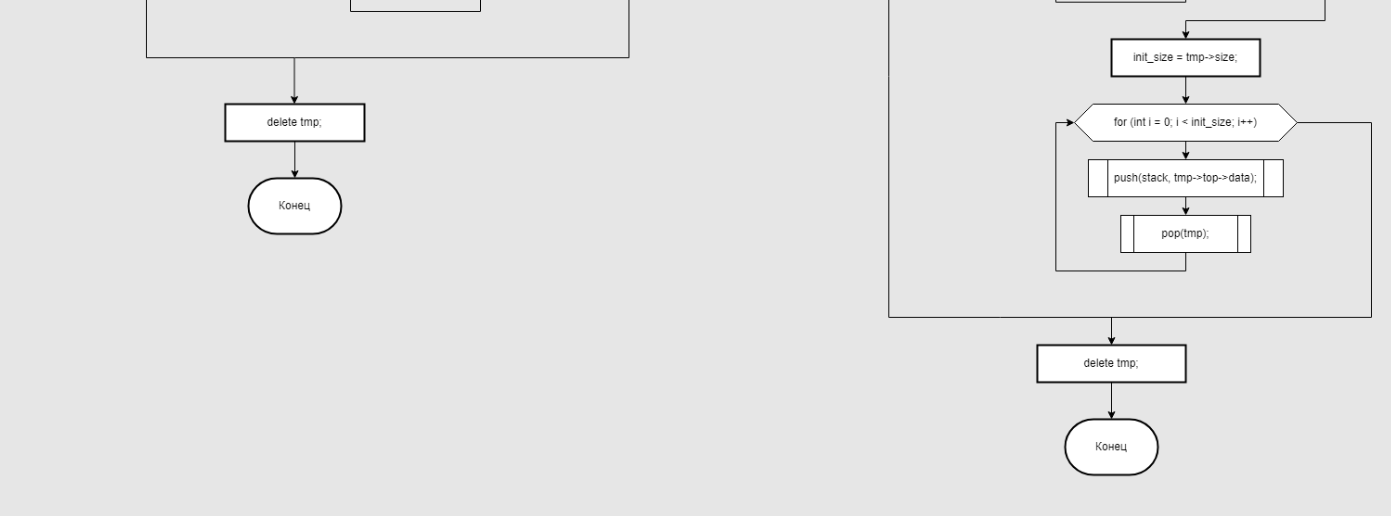
Сформировать стек, добавить в него k элементов, удалить из него k элементов, реализовать поиск элемента по его значению.

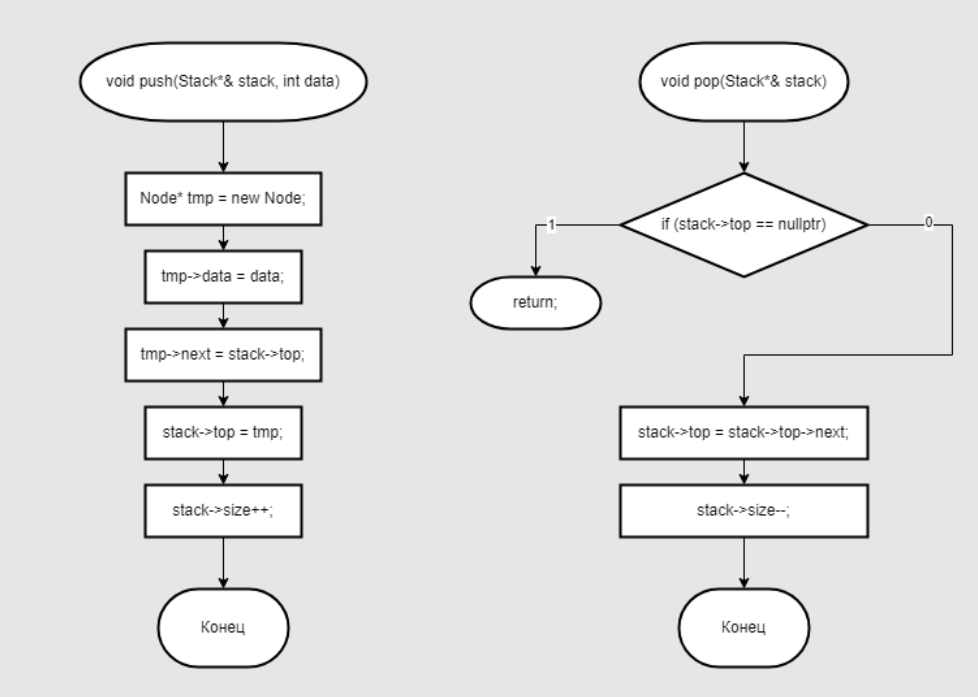
**2. Блок-схема.**

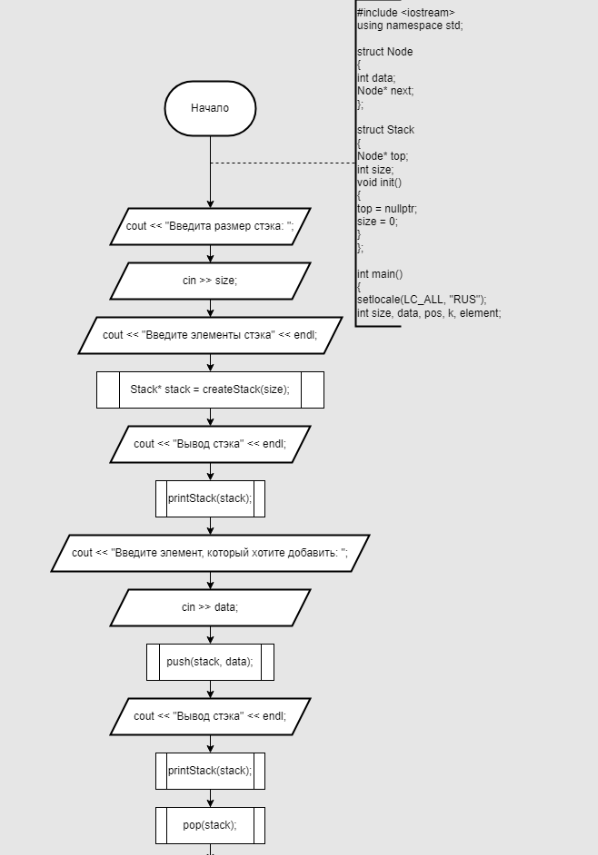


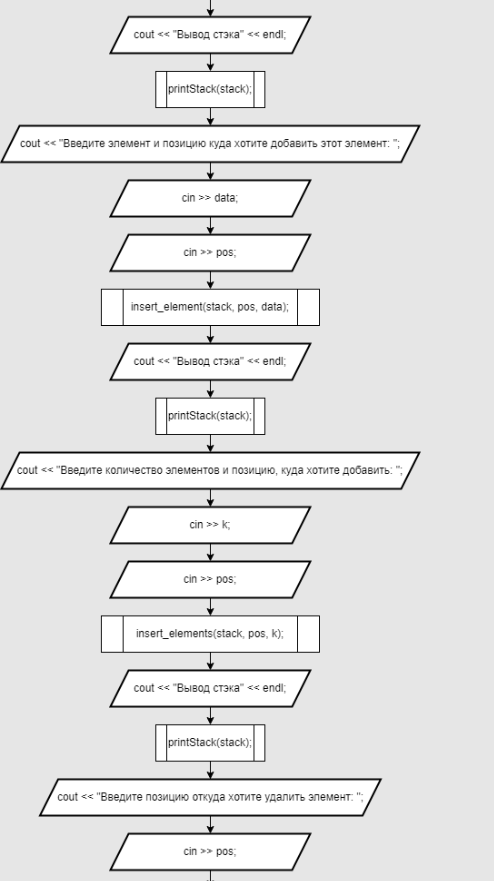


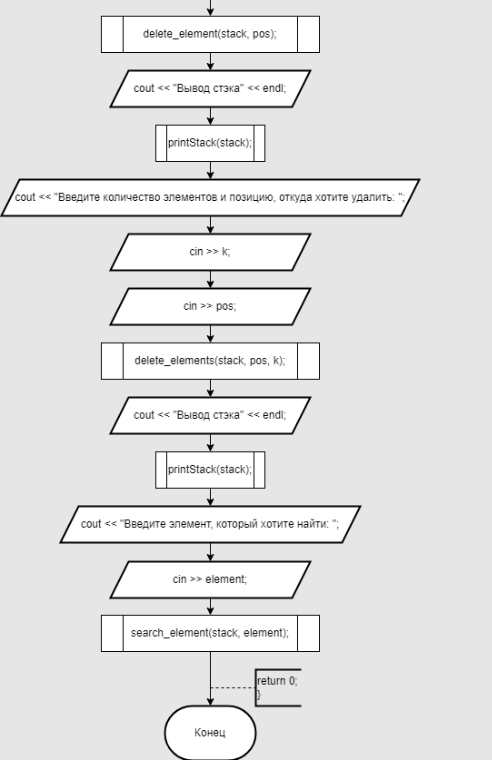












**3. Код.**

#include <iostream>

using namespace std;

struct Node

{

int data;

Node\* next;

};

struct Stack

{

Node\* top;

int size;

void init()

{

top = nullptr;

size = 0;

}

};

void push(Stack\*& stack, int data)

{

Node\* tmp = new Node;

tmp->data = data;

tmp->next = stack->top;

stack->top = tmp;

stack->size++;

}

void pop(Stack\*& stack)

{

if (stack->top == nullptr)

{

return;

}

stack->top = stack->top->next;

stack->size--;

}

void insert\_element(Stack\*& stack, int pos, int data)

{

Stack\* tmp = new Stack;

tmp->init();

int init\_size = stack->size;

if (pos == 1)

{

push(stack, data);

}

else

{

for (int i = 0; i <= init\_size - pos; i++)

{

push(tmp, stack->top->data);

pop(stack);

}

push(stack, data);

init\_size = tmp->size;

for (int i = 0; i < init\_size; i++)

{

push(stack, tmp->top->data);

pop(tmp);

}

}

delete tmp;

}

void insert\_elements(Stack\*& stack, int pos, int k)

{

Stack\* tmp = new Stack;

tmp->init();

int data;

int init\_size = stack->size;

cout << "Введите элементы, которые хотите добавить: ";

if (pos == 1)

{

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

{

cin >> data;

push(stack, data);

}

}

else

{

for (int i = 0; i <= init\_size - pos; i++)

{

push(tmp, stack->top->data);

pop(stack);

}

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

{

cin >> data;

push(stack, data);

}

init\_size = tmp->size;

for (int i = 0; i < init\_size; i++)

{

push(stack, tmp->top->data);

pop(tmp);

}

}

delete tmp;

}

void delete\_element(Stack\*& stack, int pos)

{

Stack\* tmp = new Stack;

tmp->init();

int init\_size = stack->size;

if (pos == 1)

{

pop(stack);

}

else

{

for (int i = 0; i < init\_size - pos; i++)

{

push(tmp, stack->top->data);

pop(stack);

}

pop(stack);

init\_size = tmp->size;

for (int i = 0; i < init\_size; i++)

{

push(stack, tmp->top->data);

pop(tmp);

}

}

delete tmp;

}

void delete\_elements(Stack\*& stack, int pos, int k)

{

Stack\* tmp = new Stack;

tmp->init();

int init\_size = stack->size;

if (pos == 1)

{

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

{

pop(stack);

}

}

else

{

for (int i = 0; i < init\_size - pos; i++)

{

push(tmp, stack->top->data);

pop(stack);

}

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

{

pop(stack);

}

init\_size = tmp->size;

for (int i = 0; i < init\_size; i++)

{

push(stack, tmp->top->data);

pop(tmp);

}

}

delete tmp;

}

void search\_element(Stack\*& stack, int element)

{

Stack\* curr = new Stack;

curr->init();

while (stack->top != nullptr)

{

if (stack->top->data == element)

{

cout << "Элемент найден." << endl;

return;

}

else

{

push(curr, stack->top->data);

pop(stack);

}

}

cout << "Элемент не найден." << endl;

}

Stack\* createStack(int n)

{

Stack\* stack = new Stack;

stack->init();

int data;

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

{

cin >> data;

push(stack, data);

}

return stack;

}

void printStack(Stack\*& stack)

{

Node\* curr = stack->top;

for (int i = 0; i < stack->size; i++)

{

cout << curr->data << ' ';

curr = curr->next;

}

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "RUS");

int size, data, pos, k, element;

cout << "Введита размер стека: ";

cin >> size;

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

Stack\* stack = createStack(size);

cout << "Вывод стека:" << endl;

printStack(stack);

cout << "Введите элемент, который хотите добавить: ";

cin >> data;

push(stack, data);

cout << "Вывод стека:" << endl;

printStack(stack);

pop(stack);

cout << "Вывод стека:" << endl;

printStack(stack);

cout << "Введите элемент и позицию куда хотите добавить этот элемент: ";

cin >> data;

cin >> pos;

insert\_element(stack, pos, data);

cout << "Вывод стека:" << endl;

printStack(stack);

cout << "Введите количество элементов и позицию, куда хотите добавить: ";

cin >> k;

cin >> pos;

insert\_elements(stack, pos, k);

cout << "Вывод стека:" << endl;

printStack(stack);

cout << "Введите позицию откуда хотите удалить элемент: ";

cin >> pos;

delete\_element(stack, pos);

cout << "Вывод стека:" << endl;

printStack(stack);

cout << "Введите количество элементов и позицию, откуда хотите удалить: ";

cin >> k;

cin >> pos;

delete\_elements(stack, pos, k);

cout << "Вывод стека:" << endl;

printStack(stack);

cout << "Введите элемент, который хотите найти: ";

cin >> element;

search\_element(stack, element);

return 0;

}

**4. Скриншот вывода.**

