Министерство высшего образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

**лабораторная работа №11 - стэк**

Дисциплина: основы алгоритмизации и программирования

Выполнил:

Студент группы РИС-22-1б

Баженов Тимофей

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

**Пермь, 2023**

**1. Постановка задачи**

Создать функции для работы с стэком

**2. Анализ задачи**

1. Структура узла (элемента стэка) содержит данные типа int и указатель на следующий элемент. Структура стэка содержит только указатель на головной элемент (верхушка стэка) – head\_node.

2. Функция print\_stack выводит на экран все элементы стэка.

3. Функция pop\_front удаляет элемент с верхушки стэка, перенося указатель head\_node на следующий элемент стэка.

4. Функция push\_front добавляет элемент на верхушку стэка. Если стэк был пустой – то новый элемент становиться head\_node и ссылается на nullptr. Если элементы уже были, то новый узел будет указывать на head\_node, а потом сам становится head\_node.

5. Функция del начиная с head\_node перебирает элементы стэка и, если элемент четный – то удаляет, если нечетный – переносит в вспомогательный стэк.

6. Функция swap переносит все элементы из одного стэка в другой.

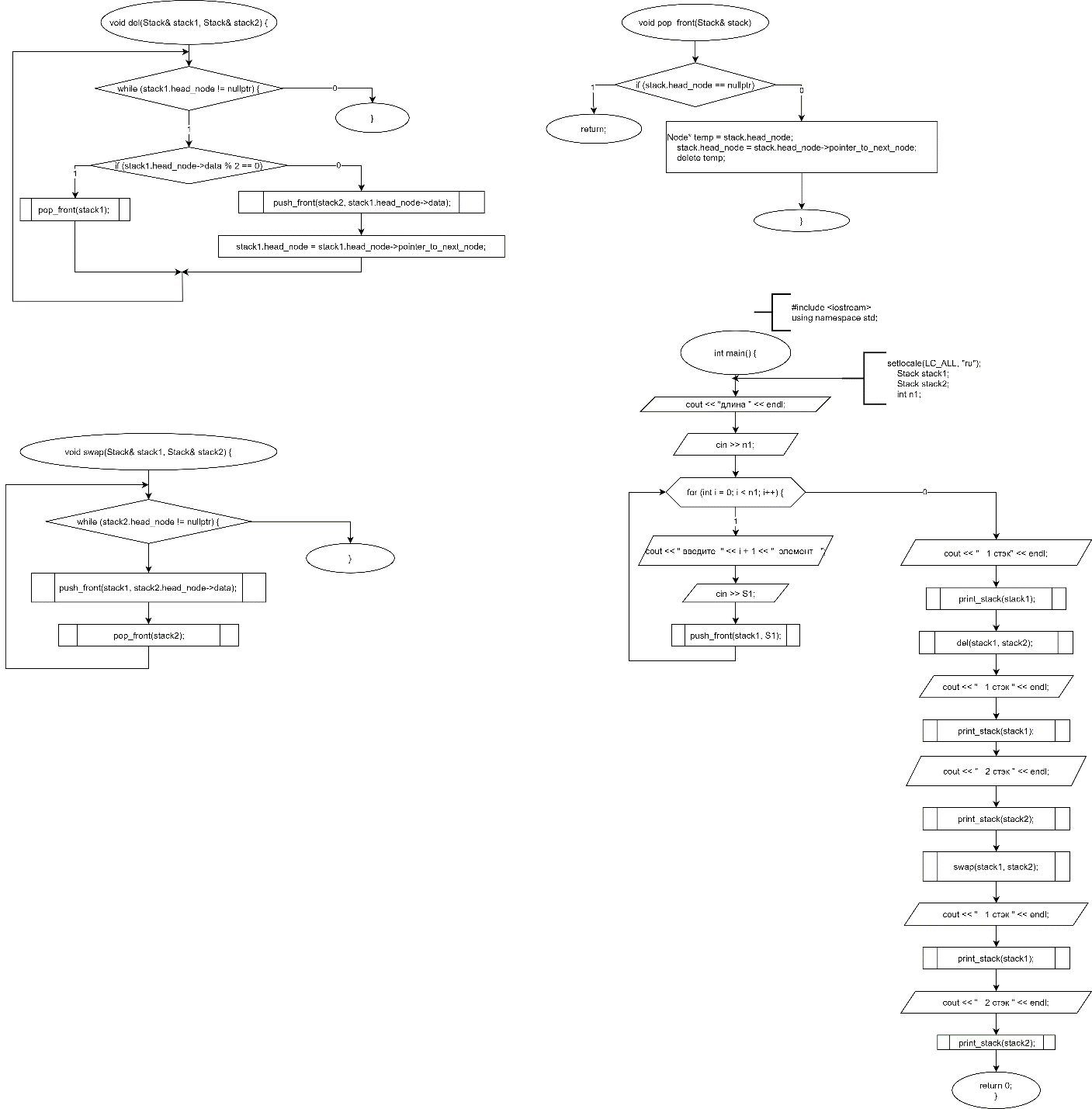
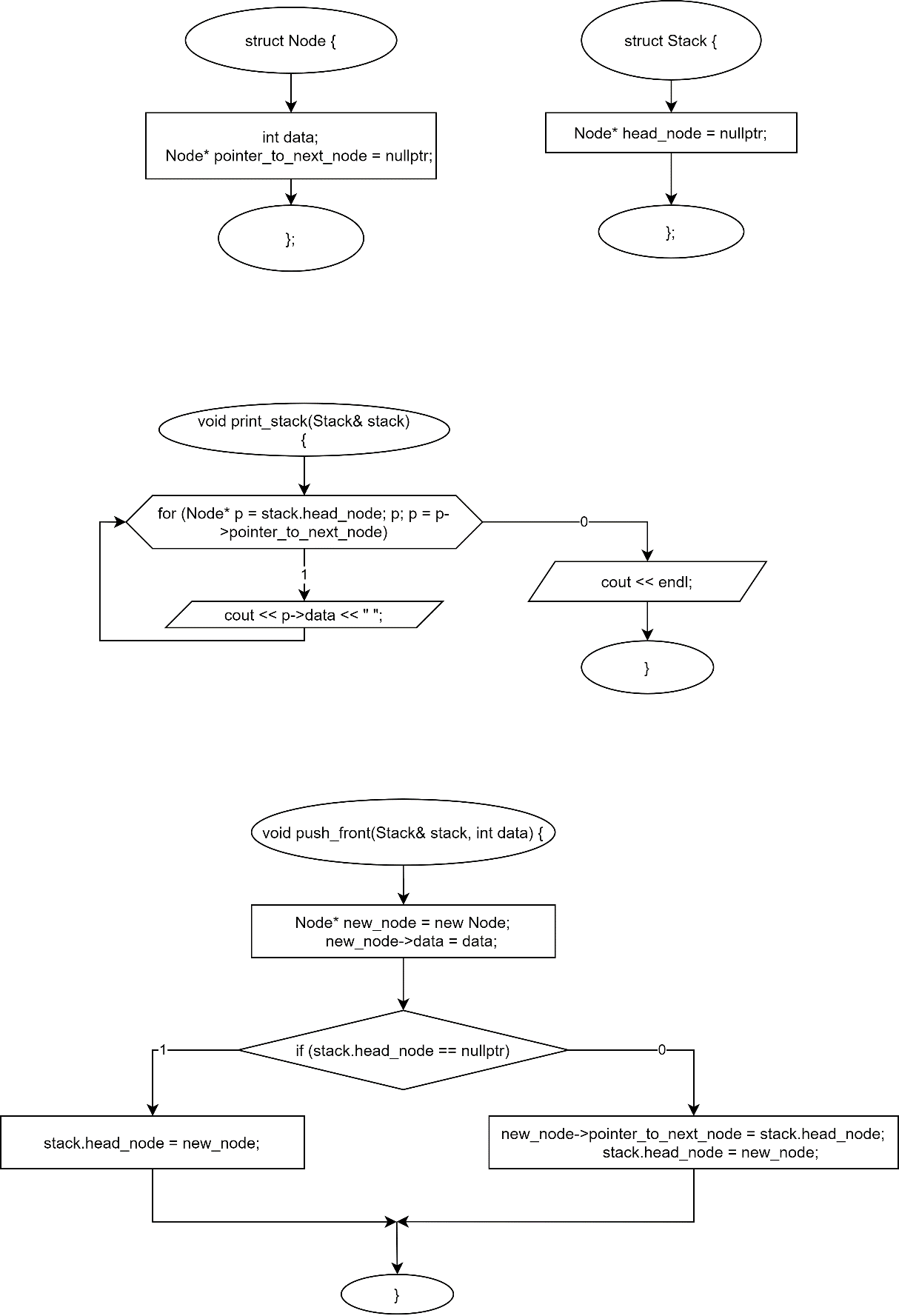
7. Функция int main(). Создаются два элементы структуры Stack – stack1 и stack2 (вспомогательный).

Пользователь вводит длину стэка и циклом for стэк stack1 заполняется значениями, которые вводит пользователь, вызывая функцию push\_back.

Сначала выводится stack1 функцией print\_stack.

Затем вызывается функция del. Первый стэк остается пустым, а во втором стэка только нечетные значения, но в обратном порядке. Оба стэка выводятся на экран.

Функцией swap все элементы из вспомогательного стэка переносятся в первый stack1. Значения снова в нужном порядке. Второй стэк становится пустым. Оба стэка выводятся на экран.

**3. Блок-схема**

**4. Исходный код**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

struct Point {

    string value;

    Point\* next = nullptr;

};

struct Stack {

    Point\* head = nullptr;

};

void popfront(Stack& stack){

    if (stack.head == nullptr) return;

    Point\* temp = stack.head;

    stack.head = stack.head->next;

    delete temp;

}

void printst(Stack& stack){

    for (Point\* temp = stack.head; temp; temp = temp->next){

        cout << temp->value << "\n";

    }

    cout << '\n';

}

void pushfront(Stack& stack, string value) {

    Point\* new\_node = new Point;

    new\_node->value = value;

    if (stack.head == nullptr)

        stack.head = new\_node;

    else {

        new\_node->next = stack.head;

        stack.head = new\_node;

    }

}

void addstack(Stack& stack){

    ofstream out("file.txt");

    for (Point\* temp = stack.head; temp; temp = temp->next){

        out << temp->value << "\n";

    }

    out << '\n';

    out.close();

}

void fromstack(Stack& stack, int size){

    string buf;

    ifstream in("file3.txt");

    for (int i = 0; i < size; i++){

        getline(in, buf);

        pushfront(stack, buf);

    }

    in.close();

}

int main() {

    Stack stack;

    int size;

    string str;

    cout << "Количество элементов стека? " << '\n';

    cin >> size;

    cin.ignore();

    for (int i = 0; i < size; i++) {

        cout << "введите " << i + 1 << " элемент: ";

        getline(cin, str);

        pushfront(stack, str);

    }

    cout << "стек" << '\n';

    printst(stack);

    getline(cin, str);

    pushfront(stack, str);

    printst(stack);

    popfront(stack);

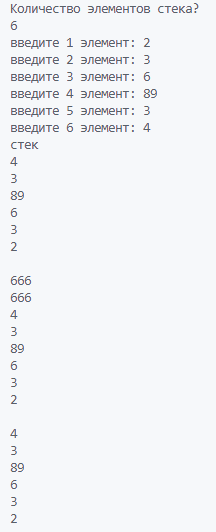
    printst(stack);

    addstack(stack);

    return 0;

}

**5. Вывод в консоль**

****