



**«Московский государственный технический университет  
имени Н.Э. Баумана»  
(национальный исследовательский университет)  
(МГТУ им. Н.Э. Баумана)**

---

Федеральное государственное бюджетное образовательное учреждение высшего образования

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**О т ч е т**

**по домашнему заданию № 3**

**Название домашней работы:** Динамические структуры данных и файловая система.

**Дисциплина:** Алгоритмизация и программирование

Студент гр. ИУ6-15Б

(Подпись, дата)

В.А. Бирюков

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.А. Веселовский

(И.О. Фамилия)

Москва, 2024

## Вариант 2

### Часть 1

**Задание:** Создать список по типу стека из вводимых целых чисел. Удалить числа, кратные 5. При завершении программы освободить динамическую память.

Напишем код программы на языке C++

```
#include <iostream>
#include <Windows.h>

struct Node {
    int data;
    Node* next;
};

void addEl(Node*& top, int value) {
    Node* newNode = new Node();
    newNode->data = value;
    newNode->next = top;
    top = newNode;
}

void deleteKrat5(Node*& top) {
    if (top == nullptr) {
        std::cout << "Пустой список" << std::endl;
        return;
    }

    while (top != nullptr && top->data % 5 == 0) {
        Node* temp = top;
        top = top->next;
        delete temp;
    }

    Node* current = top;
    while (current->next != nullptr) {
        if (current->next->data % 5 == 0) {
            Node* temp = current->next;
            current->next = temp->next;
            delete temp;
        } else {
            current = current->next;
        }
    }
}

void printList(Node* top) {
```

```

    if (top == nullptr) {
        std::cout << "Список пустой" << std::endl;
    }
    while (top != nullptr) {
        std::cout << top->data << " ";
        top = top->next;
    }
}

void deleteList(Node*& top) {
    while (top != nullptr) {
        Node* temp = top -> next;
        delete top;
        top = temp;
    }
}

int main(){
    SetConsoleOutputCP(CP_UTF8);
    Node* top = nullptr;
    bool flag = true;
    std::string input;
    std::cout << "Введите числа в список (end для завершения ввода)" <<
std::endl;
    while (flag) {
        std::cin >> input;
        if (input == "end") {
            flag = false;
        } else {
            int value = std::stoi(input);
            addEl(top, value);
        }
    }
    std::cout << "Введённый список" << std::endl;
    printList(top);
    std::cout << std::endl << "Список без чисел, кратных 5" << std::endl;
    deleteKrat5(top);
    printList(top);
    deleteList(top);
    return 0;
}

```

Запустим программу, чтобы проверить корректность её работы

```
Введите числа в список (end для завершения ввода)
-123 39 0 35 71 50 -15 22 103128038 98
end
Введённый список
98 103128038 22 -15 50 71 35 0 39 -123
Список без чисел, кратных 5
98 103128038 22 71 39 -123
```

Рисунок 1 – результат работы программы

Видим, что программа корректно работает на введённых числах

Составим схемы алгоритмов для этой программы

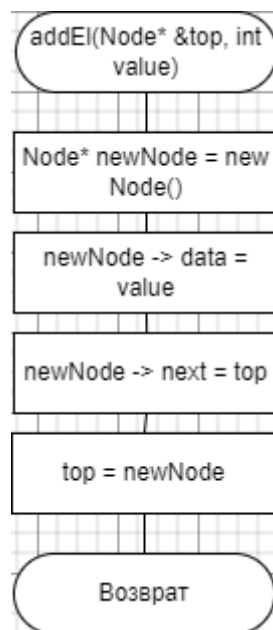


Рисунок 2 – схема алгоритма процедуры для добавления элементов в список

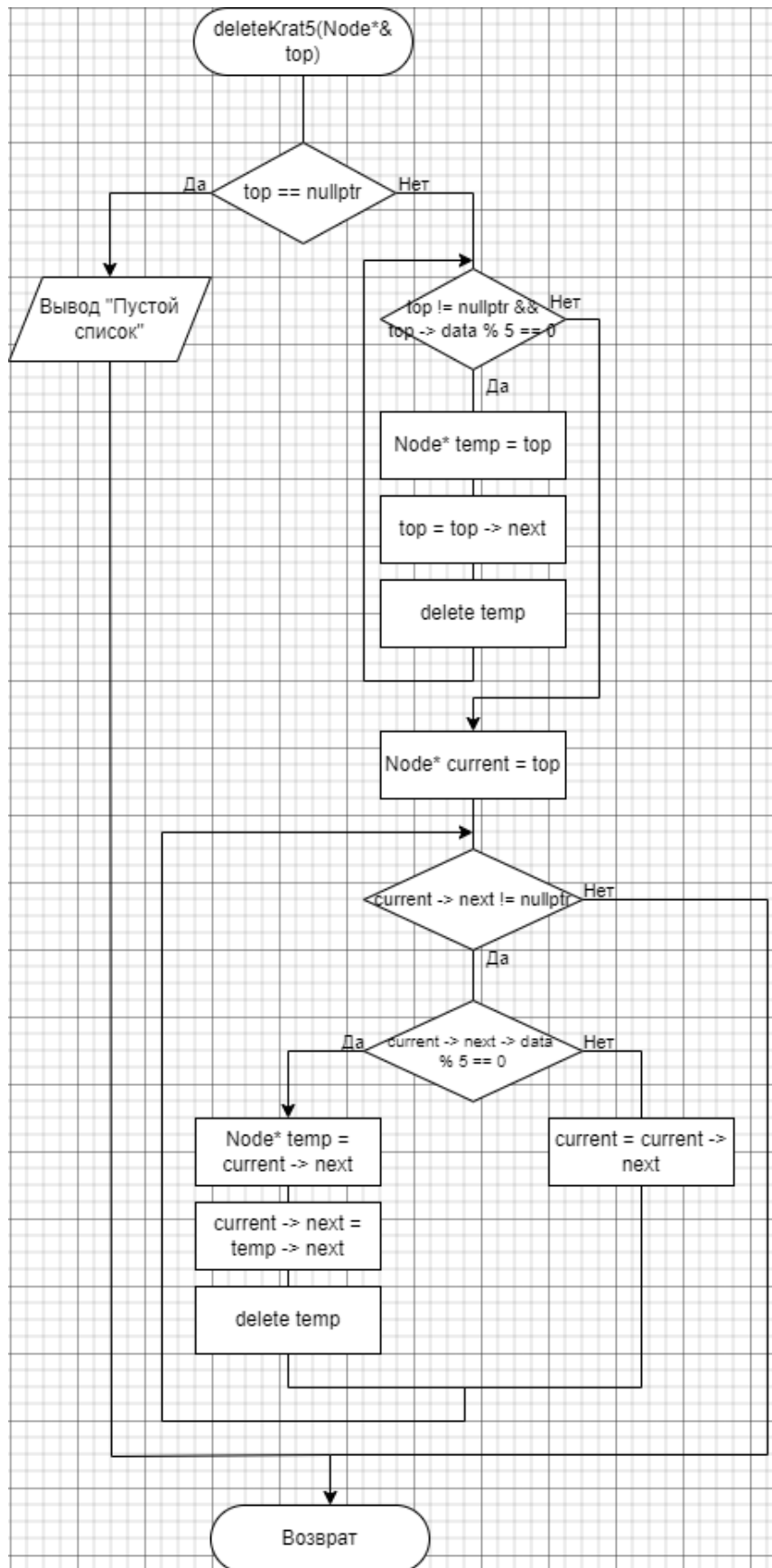


Рисунок 3 – схема алгоритма процедуры для удаления элементов списка, кратных 5

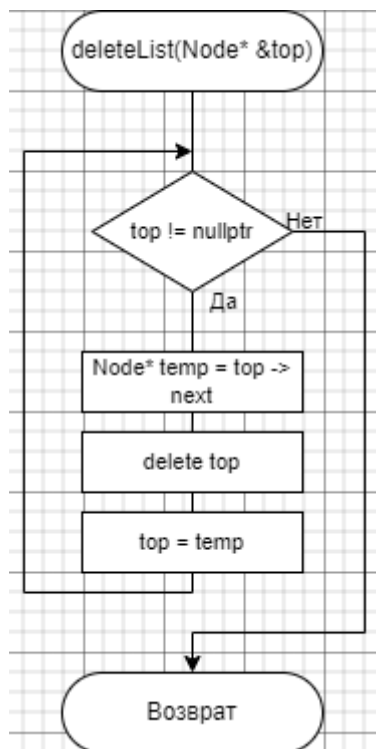


Рисунок 4 – схема алгоритма процедуры для освобождения памяти

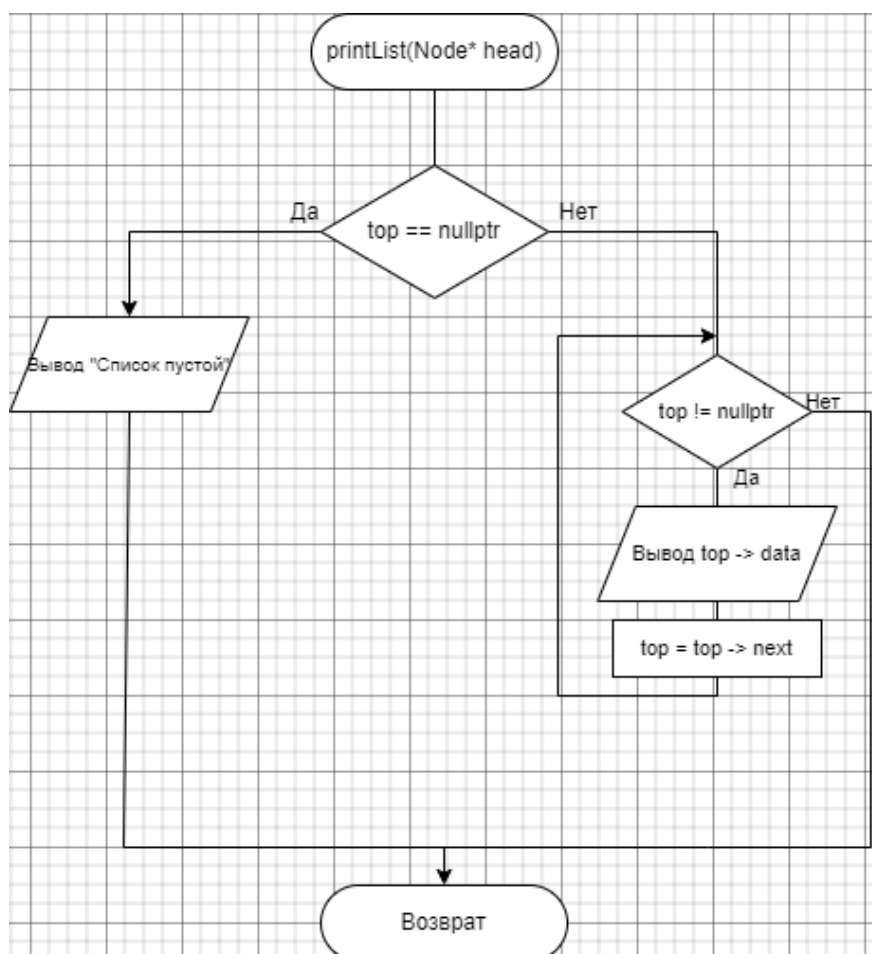


Рисунок 5 – схема алгоритма процедуры для вывода списка

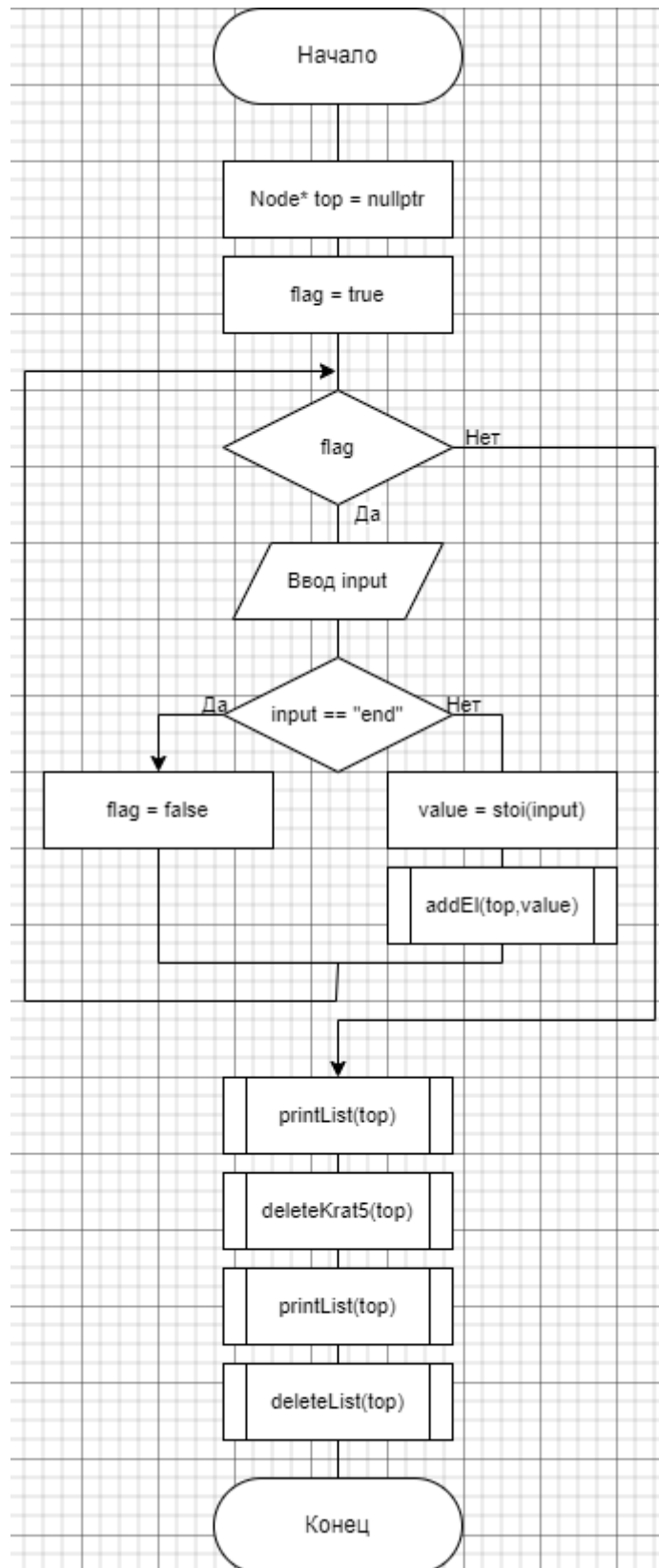


Рисунок 6 – схема алгоритма основной программы

## Часть 2

**Задание:** Создать файл F, состоящий из целых чисел. Переписать из файла F в файл G все нечетные числа. Вывести на экран содержимое обоих файлов.

Напишем код программы на языке C++

```
#include <iostream>
#include <fstream>

int main()
{
    std::string str;
    std::ifstream f1("F.txt");
    std::ofstream f2("G.txt");
    std::cout << "File F: " << std::endl;
    while(std::getline(f1, str)) {
        std::cout << str << " ";
        if (stoi(str) % 2 != 0) {
            f2 << str << std::endl;
        }
    }
    std::cout << std::endl;
    f2.seekg(0);
    std::cout << "File G: " << std::endl;
    while(std::getline(f2, str)) {
        std::cout << str << " ";
    }
    return 0;
}
```

Запустим программу, чтобы убедиться в правильности работы программы

```
File F:
3 12 310 123 122 10 1 5784 91 -2137 1738 -12382317 1929 8479 -12390 0 19283 32 987
File G:
3 123 1 91 -2137 -12382317 1929 8479 19283 987
```

Рисунок 7 – результат работы программы



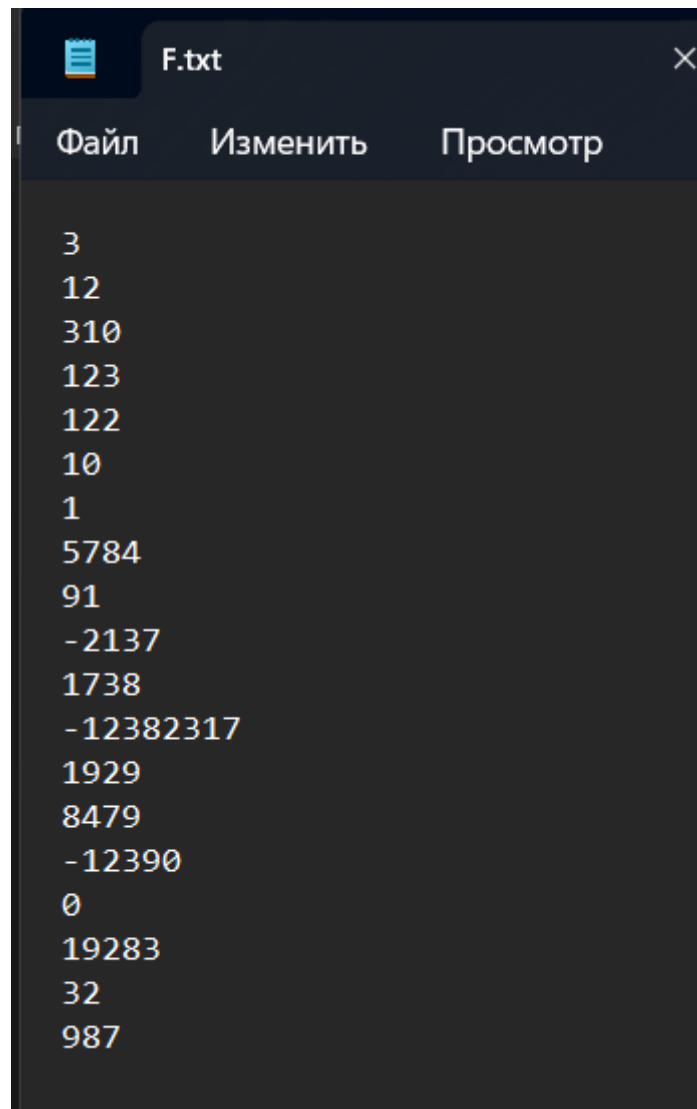


Рисунок 8 – содержимое файла F.txt

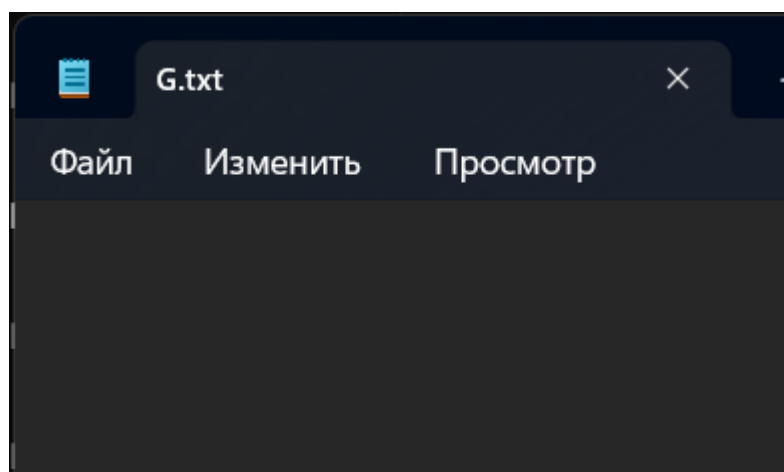


Рисунок 9 – содержимое файла G.txt до запуска программы

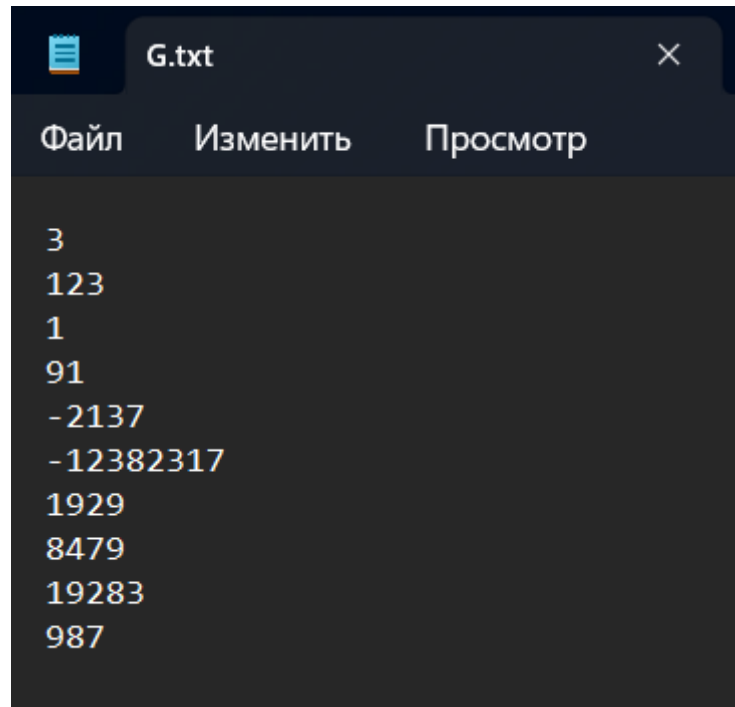


Рисунок 10 – содержимое файла G.txt после запуска программы

Видим, что программа корректно работает

Составим схему алгоритма для этой программы

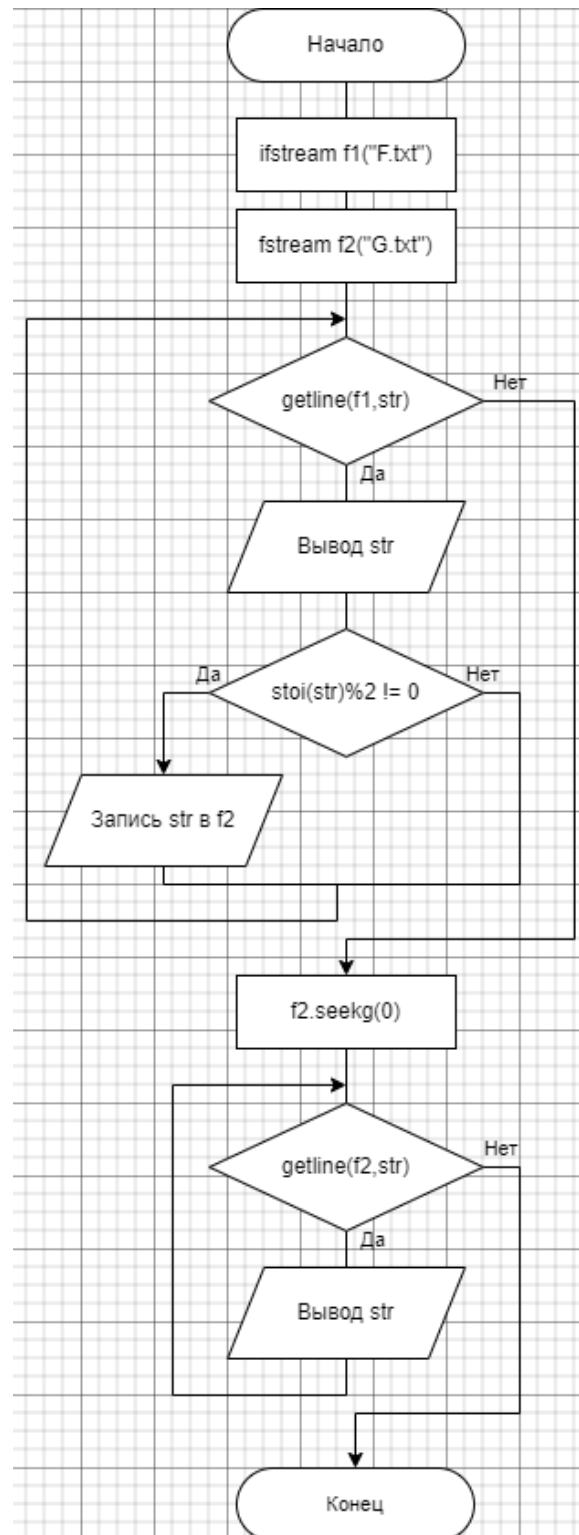


Рисунок 11 – схема алгоритма программы

**Вывод:** в ходе выполнения домашнего задания я получил базовые навыки работы с списковыми структурами и с файлами в C++