ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| к.т.н, доц.. |  |  |  | А.В. Туманова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка данных в файлах |
| по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  | 17.03.22 |  | Д.А. Кузнецов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. **Цель работы**

Целью работы является изучение файловых операций ввода/вывода.

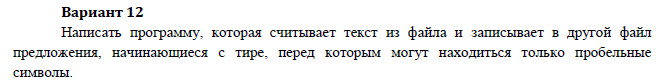
1. **Задание**

Работа выполняется в двух вариантах:

1. С использованием библиотек потокового ввода-вывода (<fstream>).

2. С использованием библиотек стандартного ввода-вывода (<cstdio>).

Примечание: если использовались операторы динамического выделения памяти, то следует вставить дополнительный код, обнаруживающий утечки памяти.



1. **Описание созданных функций**

**Имя:** main

**Назначение**: открытие и закрытие входного и выходного файлов и промежуточные вычисления

**Входные данные:** ifstream: in – входной файл

ofstream: out – выходной файл

**Побочный эффект:** отсутствует

**Тестовые данные:**

**input.txt**

**“**

Привет, как дела? - Да никак, номально.

Недавно закончил помогать маме. - Вот сижу, учусь... - Получается?

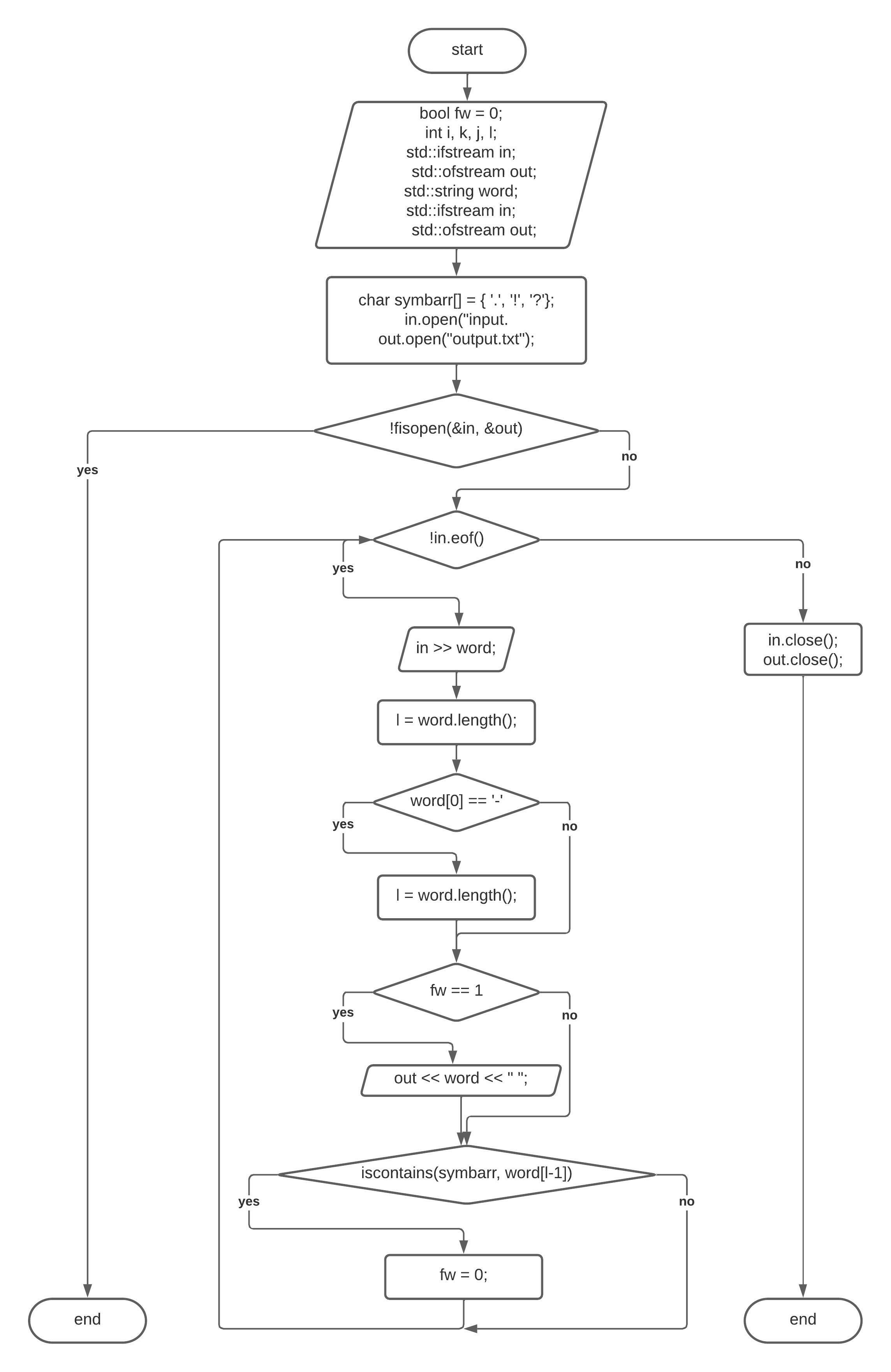
- пквапвп

12345

**“**

**Прототип:** int main()

**Блок-схема:**



**Текст программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | **int** **main**() {  setlocale(LC\_ALL, "rus");  **char** symbarr[] = { '.', '!', '?'};  std::ifstream in;  std::ofstream out;    in.open("input.txt");  out.open("output.txt");  **if** (!fisopen(&in, &out)) {  **return** **0**;  }  /\*  Проверяем строку на заданный шаблон, если предложение начинается с тире,  то мы записываем в файл output.txt  \*/  filter(&in, &out, symbarr);  in.close();  out.close();  **return** **0**;  } | |  |  |  |
|  |  |

**Имя:** fisopen

**Назначение**: проверка на корректность открытия файла

**Входные данные:** ifstream\*: in – входной файл

ofstream\*: out – выходной файл

bool: f = 1;

**Побочный эффект:** отсутствует

**Тестовые данные:**

**input.txt**

**“**

Привет, как дела? - Да никак, номально.

Недавно закончил помогать маме. - Вот сижу, учусь... - Получается?

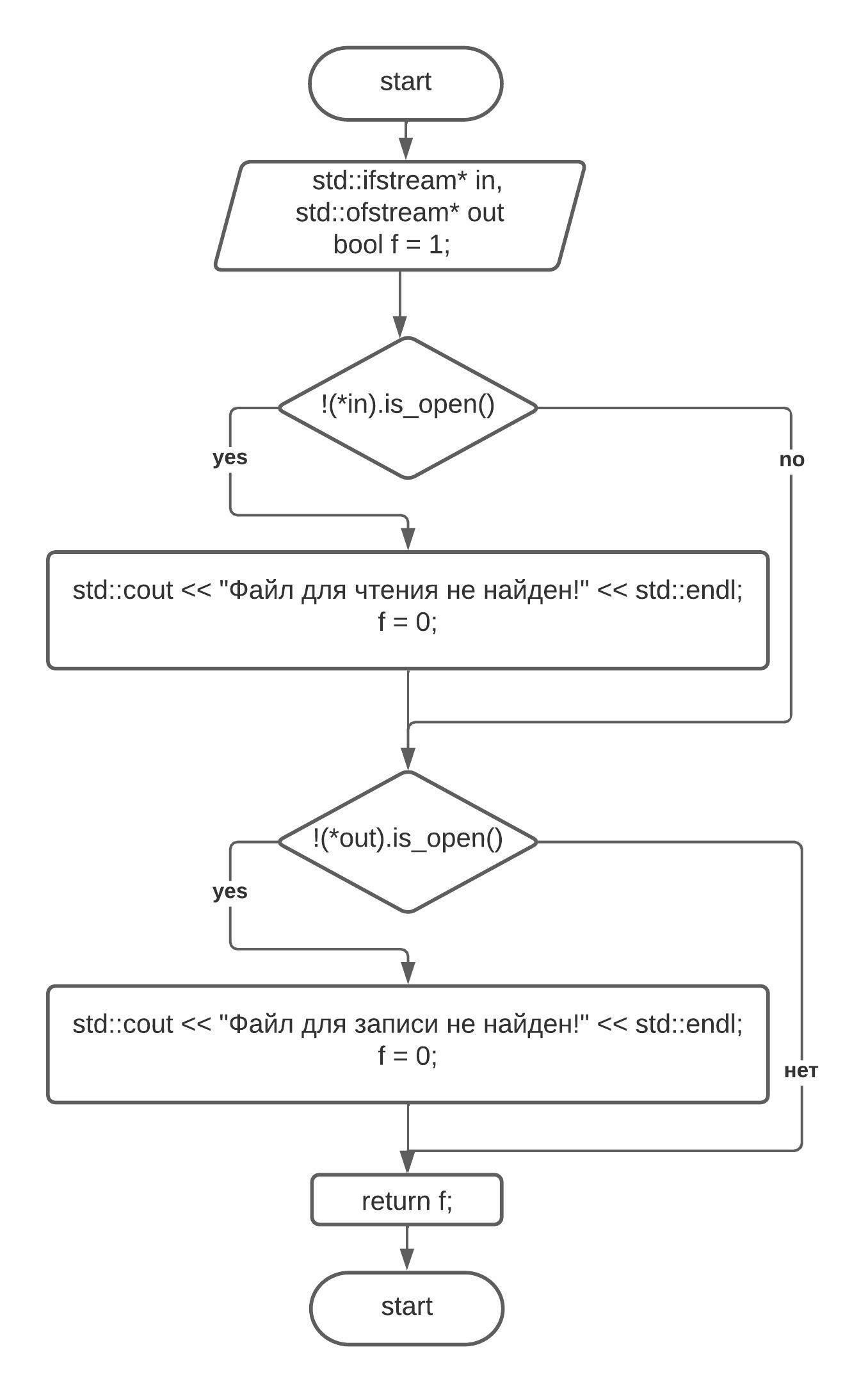
- пквапвп

12345

**“**

**Прототип**: bool fisopen(std::ifstream\* in, std::ofstream\* out)

**Блок-схема:**



**Текст программы:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | **bool** **fisopen**(std::ifstream\* in, std::ofstream\* out) {  **bool** f = **1**;  **if** (!(\*in).is\_open()) {  std::cout << "Файл для чтения не найден!" << std::endl;  f = **0**;  }  **if** (!(\*out).is\_open()) {  std::cout << "Файл для записи не найден!" << std::endl;  f = **0**;  }  **return** f;  } | |  |  |  |

**Имя:** iscontains

**Назначение**: Проверка наличия символа в массиве символов

**Входные данные:** char: symbarr[] – массив символов

char: symb – символ для проверки

bool: f = 0 - флаг

int: i - счётчик

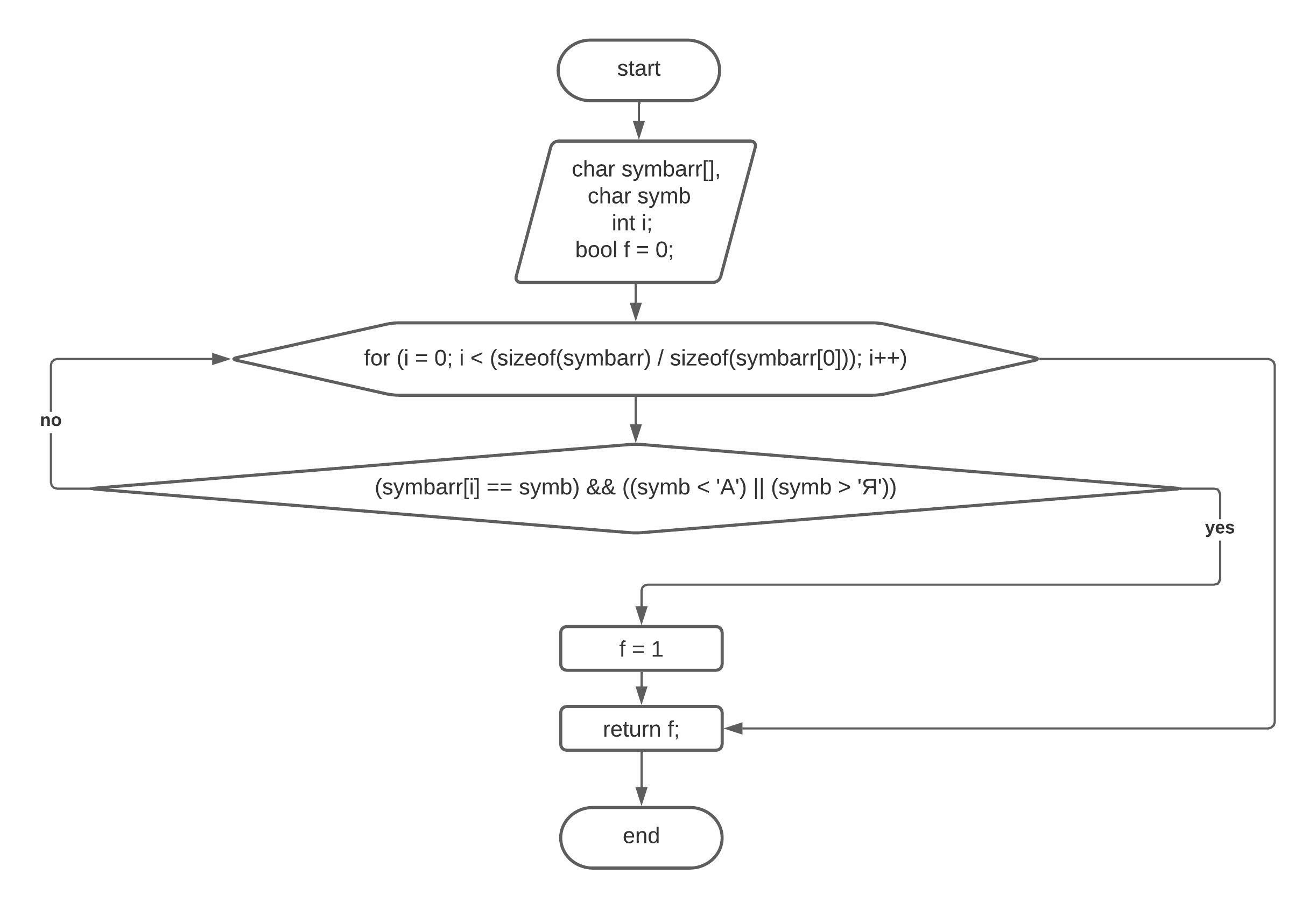
**Тестовые данные:**

char symbarr[] **=** { '.', '!', '?' };

symb = ‘a’;

**Прототип**: bool iscontains(char symbarr[], char symb)

**Блок-схема:**



**Текст программы:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **bool** **iscontains**(**char** symbarr[], **char** symb) {  **int** i;  **bool** f = **0**;  **for** (i = **0**; i < (**sizeof**(symbarr) / **sizeof**(symbarr[**0**])); i++) {  // условие (symb < 'А') || (symb > 'Я') для отсечения  Больших русских букв, так как их отрицательные  // коды пересекаются со спец символами из массива  **if** ((symbarr[i] == symb) && ((symb < 'А') || (symb > 'Я'))) {  f = **1**;  **break**;  }  }  **return** f;  } | | |  |  |  |

**Имя:** filter

**Назначение**: запись в выходной файл только тех предложений, которые удовлетворяют условию

**Входные данные:** char: symbarr[] – массив символов

ifstream\*: in – входной файл

ofstream\*: out – выходной файл

string: word – считываемое слово

bool: fw = 0 - флаг

int: l – длина считываемого слова

**Побочный эффект:** отсутствует

**Тестовые данные:**

char symbarr[] **=** { '.', '!', '?' };

fstream\* in = &in;

ofstream\* out = &out;

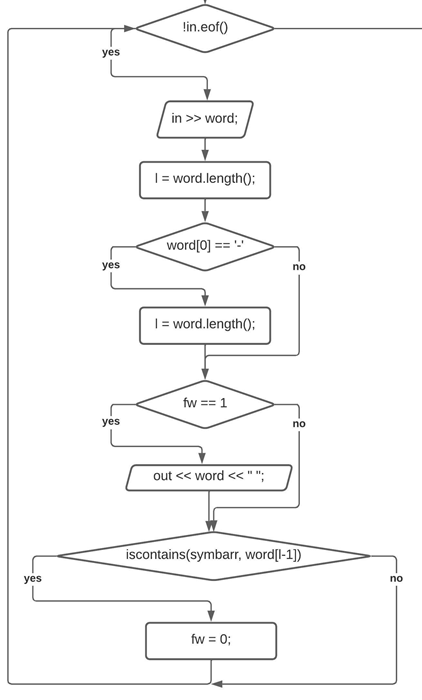
string word = “ ”;

bool fw = 0;

int: l = 0;

**Прототип**: void filter(std::ifstream\* in, std::ofstream\* out, char symbarr[])

**Блок-схема:**

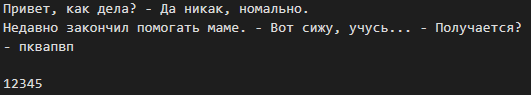
****

**Текст программы:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | **void** **filter**(std::ifstream\* in, std::ofstream\* out, **char** symbarr[]) {  **int** l;  std::string word;  **bool** fw = **0**; // flag word (Показатель того, что текущее предложение начинается с тире)  **while** (!(\*in).eof()) {  \*in >> word;  l = word.length();  **if** (word[**0**] == '-') {  fw = **1**;  }  **if** (fw == **1**) {  \*out << word << " ";  }  **if** (iscontains(symbarr, word[l - **1**])) {  fw = **0**;  }  } | |  | |  | | |  |  |  |

1. **Пример выполнения программы**

**input.txt**

****

**output.txt**

1. **Анализ результатов и выводы**

В ходе лабораторной работы я изучил файловые операции ввода/вывода.