

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных  
систем

## **Лабораторная работа №10.1**

по дисциплине: Основы программирования  
тема: «Обработка текстовых файлов»

Выполнил: ст. группы ПВ-201  
Машуров Дмитрий Русланович

Проверил:  
Притчин Иван Сергеевич  
Брусенцева Валентина  
Станиславовна

Белгород 2021 г.

## **Лабораторная работа № 10.1**

### **Обработка текстовых файлов**

**Цель работы:** получение навыков работы с потоками.

#### **Задания для подготовки к работе**

1. Изучить организацию работы с текстовыми и бинарными потоками.
2. Разработать алгоритм и составить программы для решения каждой из двух задач соответствующего варианта. В бинарных файлах информация хранится в машинном представлении.
3. Для бинарных файлов составить программу для создания файла.
4. Подобрать тестовые данные.

#### **Задание варианта №16**

Из данного текстового файла удалить слова, длина которых превышает данное число  $n$ .

## Выполнение:

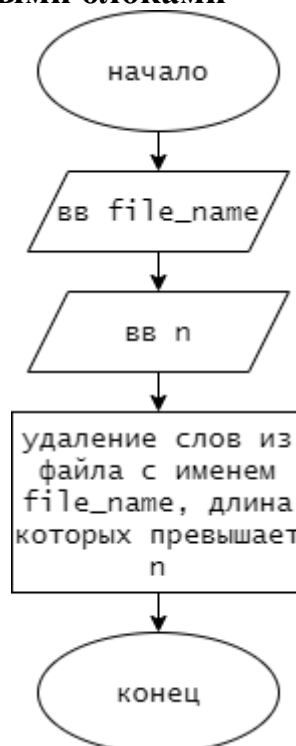
### 1. Описание алгоритма и выделение подзадач

Исходя из условия задачи, будем считывать каждое слово в буфер *buf* из входного файла, сравнивать длину считанного слова *buf* с данным числом *n* и записывать считанное слово *buf* в выходной файл, если его длина меньше, чем *n*, иначе – переходим к следующему слову

Выделение подзадач:

- 1) Удаление слов из файла, длина которых превышает заданное число
- 2) Определение длины строки

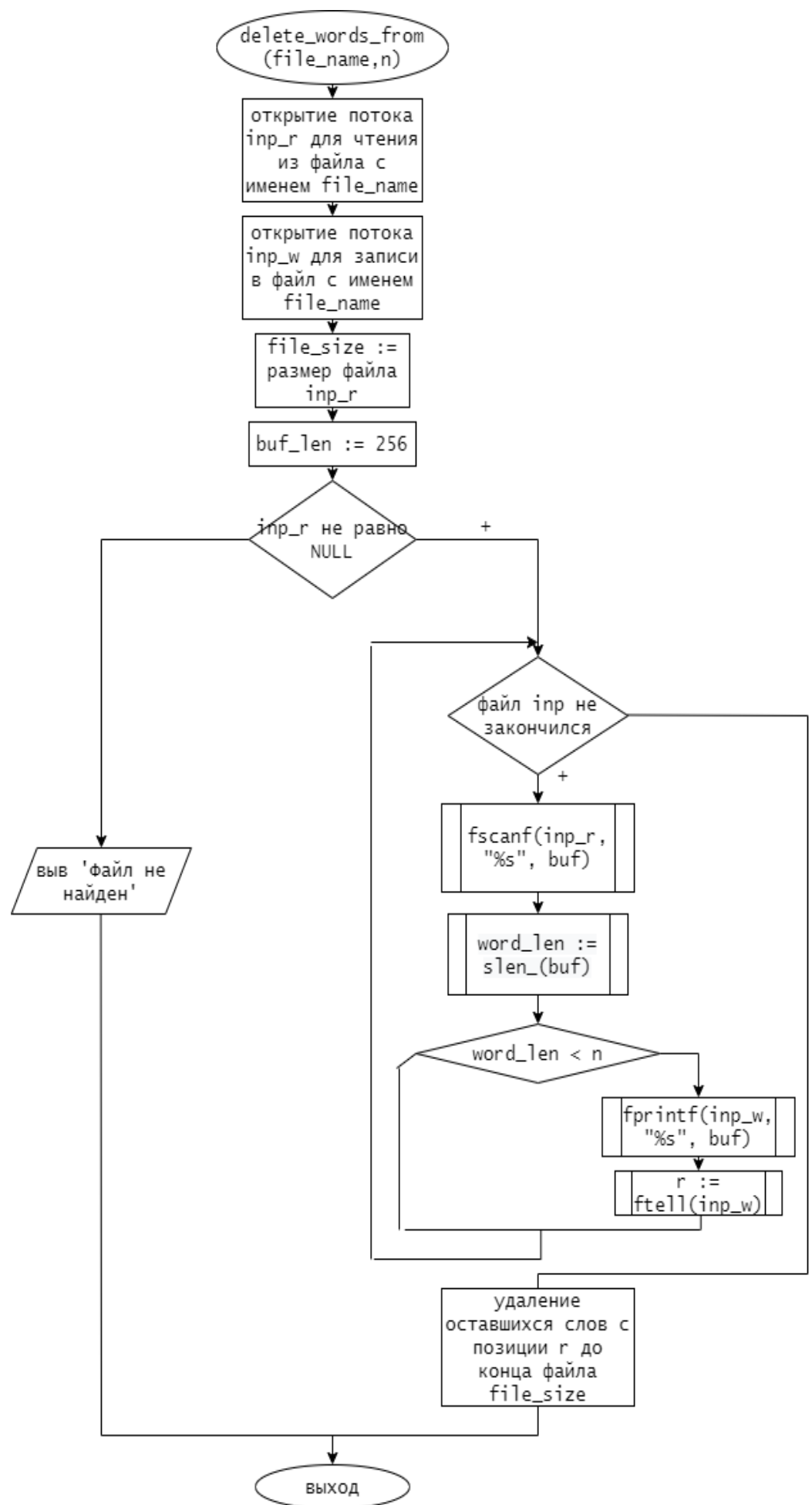
### 2. Блок-схема с укрупнёнными блоками



### 3. Спецификации подзадач

1) Удаление слов из файла, длина которых превышает заданное число

- a. Заголовок: `void delete_words_from(char *file_name, int n)`
- b. Назначение: удаляет слова из файла с именем `file_name`, длина которых превышает (или равна) `n`
- c. Блок-схема:



2) Определение длины строки

- a. Заголовок: `int slen_(char *str)`
- b. Назначение: возвращает длину строки `str`
- c. Блок-схема:



#### 4. Тестовые данные

№	Вход	Выход
1	<div><div>“inp.txt”</div><div>jotoro dio oh you're approaching me</div><div>4</div></div>	<div><div>dio oh me</div></div>
2	<div><div>“inp.txt”</div><div>a ab abc abcd abcde abcdeffffff 000000</div><div>5</div></div>	<div><div>a ab abc abcd</div></div>

#### 5. Текст программы

```
#include <stdio.h>

/* возвращает длину строки str */
size_t slen_(char *str) {
    size_t i = 0;

    while (*str != '\0') {
        i++; str++;
    }

    return i;
}

/*
 * удаляет слова из файла с именем file_name, длина
 * которых превышает (или равна) n
 */
void delete_words_from(char *file_name, size_t n) {
    FILE *inp_r = fopen(file_name, "r"),
        *inp_w = fopen(file_name, "r+");
    size_t buf_len = 256;
    long r;

    fseek(inp_r, 0, SEEK_END);
    int file_size = ftell(inp_r);
    fseek(inp_r, 0, SEEK_SET);

    if (inp_r != NULL) {
        while (feof(inp_r) == 0) {

            char buf[buf_len];
            fscanf(inp_r, "%s", buf);

            long word_len = slen_(buf);

            if (word_len < n) {
                fprintf(inp_w, "%s ", buf);
                r = ftell(inp_w);
            }
        }
    }
}
```

```

        while (r < file_size) {
            fputc(' ', inp_w);
            r++;
        }
    } else {
        perror("A file with this name wasn't found");
    }

    fclose(inp_r);
    fclose(inp_w);
}

int main() {
    printf("Input source file name\n");
    char file_name[50];
    scanf("%s", file_name);

    printf("Input number\n");
    size_t n;
    scanf("%u", &n);

    delete_words_from(file_name, n);
}

```

## 6. Результаты работы:

*Пример №1:*

```

Input source file name
inp.txt
Input number
4

```

*Пример №2:*

```

Input source file name
inp.txt
Input number
5

```