

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**  
«Программирование»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

«Строки и ввод-вывод»

*Вариант 11-2-1-4*

**Выполнил:**

Суханкулиев Мухаммет,  
студент группы N3146



(подпись)

**Проверила:**

Сыдыкова Эмилия,  
преподаватель программирования

\_\_\_\_\_  
(отметка о выполнении)

\_\_\_\_\_  
(подпись)

Санкт-Петербург  
2023 г.

## СОДЕРЖАНИЕ

Введение .....	4
1     Указатели и динамическая память .....	5
1.1   Задание.....	5
1.2   Make-файл .....	5
1.2.1   Примеры работы программы .....	6
1.3   Исходный текст программы .....	6
1.3.1   .c-файл .....	6
1.3.2   .h-файл.....	9
Заключение.....	11
Список использованных источников.....	12

## **ВВЕДЕНИЕ**

Цель работы – Разработать на языке C для ОС Linux программу, которая читает данные из стандартного потока ввода или файла, осуществляет преобразование в соответствии с вариантом лабораторной работы и выводит результат в стандартный поток вывода или файл.

Для достижения поставленной цели необходимо решить следующие задачи:

- Выполнить задание;
- Протестировать программу;
- Заархивировать папку проекта.

## 1 СТРОКИ И ВВОД-ВЫВОД

Выполнение преобразования строки в соответствии с вариантом 11-2-1-4.

### 1.1 Задание

Данные для поиска – Федеральный телефонный номер РФ Вида +7|8{()DEF{}}XXX{-}XX{-}XX, где | означает выбор одного из вариантов, {} - элемент может отсутствовать. Примеры: +7(921)3368715, 89013368584, 8(971)201-11-91.

Опции, с помощью которых задается ограничение преобразования –  
-f M

Опция -f задает номер объекта данных, начиная с которого нужно осуществлять заданное вариантом преобразование. Объекты данных нумеруются с единицы. Если опция -f не указана, то преобразование выполняется, начиная с первого найденного объекта.

-t N

Опция -t задает номер последнего объекта данных, над которым нужно осуществлять заданное вариантом преобразование. Если опция -t не указана, преобразование выполняется до последнего найденного объекта данных. При указании опции -n нумеруются только те объекты данных, которые отвечают критериям поиска, то есть располагаются на одной строке.

Разметка Markdown, используемая для выделения найденных фрагментов текста – Курсив *\*Пример\**.

Цвет для выделения найденных фрагментов текста при выводе с опцией -c – Синий (blue), Код цвета (foreground color ANSI code) 34

### 1.2 Make-файл

```
.PHONY: all clean
```

```
APP=lab3msN3146
```

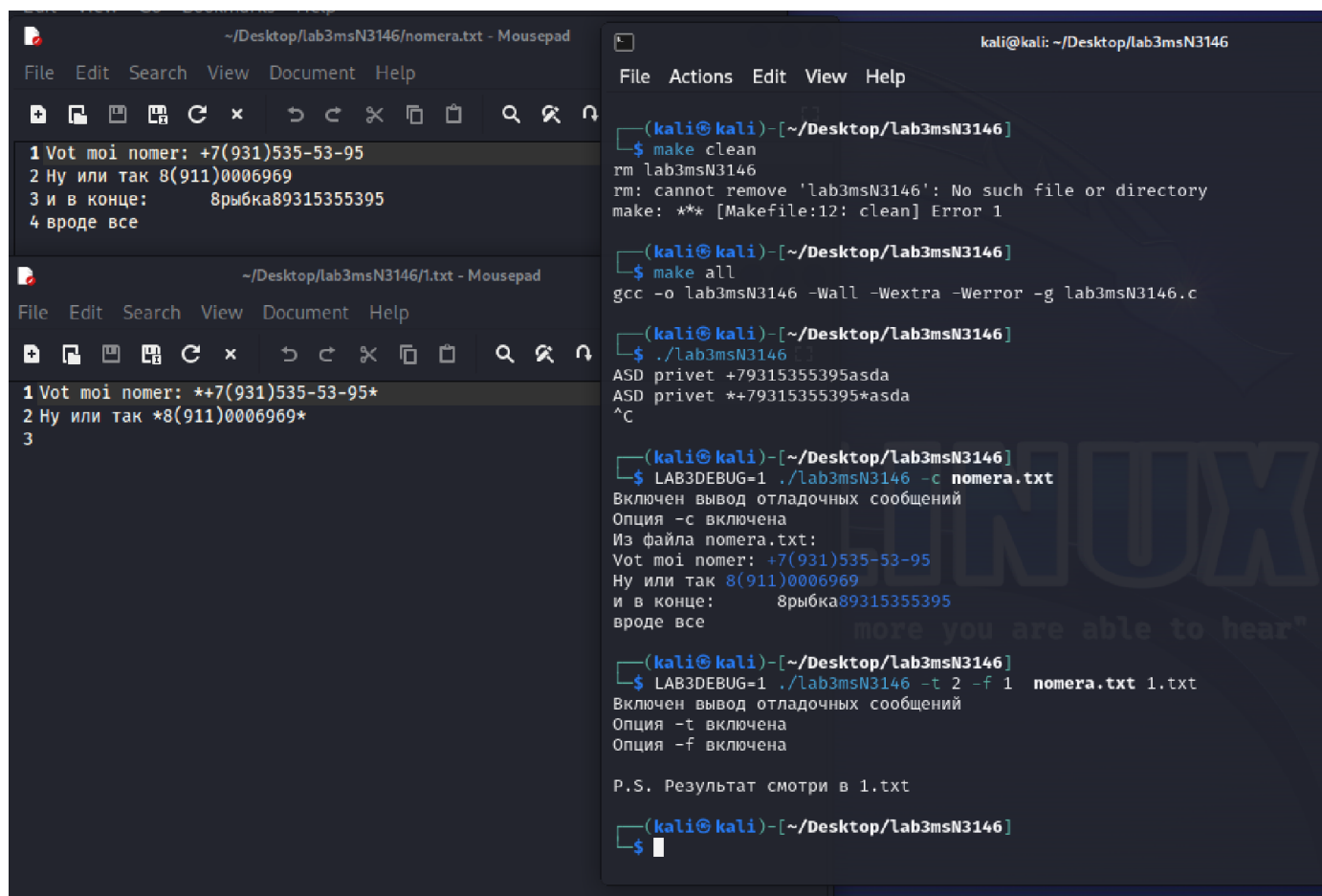
```
CFLAGS=-Wall -Wextra -Werror -g
```

```
all: $(APP)

$(APP): $(APP).c
    gcc -o $(APP) $(CFLAGS) $(APP).c

clean:
    rm $(APP)
```

## 1.2.1 Примеры работы программы



## 1.3 Исходный текст программы

### 1.3.1 .c-файл

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <regex.h>
#include <errno.h>
#include <limits.h>
#include <unistd.h>

#include "functions.h"

#define _GNU_SOURCE
```

```

int main(int argc, char *argv[]) {
    // Проверка запуска с переменной среды, включающей отладочный вывод.
    // Пример запуска с установкой переменной LAB3DEBUG в 1:
    // $ LAB3DEBUG=1 ./lab3msN3146 -с nomera.txt
    char *DEBUG = getenv("LAB3DEBUG");
    if (DEBUG) {
        fprintf(stderr, "Включен вывод отладочных сообщений\n");
    }
    if (argc > 9) {
        fprintf(stderr, "Использование: %s [опции] [имя_вход_файла
[имя_выход_файла]]\n", argv[0]);
        return EXIT_FAILURE;
    }

    int opt;
    int from = 1; // Номер первого обрабатываемого объекта (строки)
    int to = -1;  // Номер последнего обрабатываемого объекта (строки)
    int useColor = 0; // Флаг использования цвета
    int onlySameLine = 0; // Флаг ограничения поиска одной строкой
    while ((opt = getopt(argc, argv, "vf:t:cn")) != -1) {
        switch (opt) {
            case 'v':
                if (DEBUG) {
                    fprintf(stderr, "Опция -v включена\n");
                }
                fprintf(stderr, "Суханкулиев Мухаммет, гр. N3146\nВариант: 12-2-1-
4\n");
                exit(EXIT_SUCCESS);
                break;
            case 'f': // Опция "-f" для указания номера первого объекта (строки)
                from = atoi(optarg);
                if (from <= 0) {
                    fprintf(stderr, "Неверное значение параметра -f\n");
                    exit(EXIT_FAILURE);
                }
                if (DEBUG) {
                    fprintf(stderr, "Опция -f включена\n");
                }
                break;
            case 't': // Опция "-t" для указания номера последнего объекта (строки)
                if (sscanf(optarg, "%d", &to) != 1 || to <= 0) {
                    fprintf(stderr, "Неверное значение параметра -t\n");
                    exit(EXIT_FAILURE);
                }
                if (DEBUG) {
                    fprintf(stderr, "Опция -t включена\n");
                }
                break;
            case 'c': // Опция "-с" для включения использования цвета
                if (DEBUG) {
                    fprintf(stderr, "Опция -с включена\n");
                }
                useColor = 1;
                break;
            case 'n': // Опция "-n" для ограничения поиска одной строкой
                if (DEBUG) {
                    fprintf(stderr, "Опция -n включена\n");
                }
                onlySameLine = 1;
                break;
            default: // Обработка нераспознанных опций

```

```

        fprintf(stderr, "Использование: %s [-v] [-f M] [-t N] [-c] [-n]
[имя_вход_файла [имя_выход_файла]]\n", argv[0]);
        exit(EXIT_FAILURE);
    }
}
// Обработка последнего объекта (строки)
if (to == -1) {
    to = INT_MAX;
}

int numFiles = argc - optind;
// Обработка одного файла (стандартного ввода или указанного в командной строке)
if (numFiles == 0) {
    processAndPrintPhoneNumbers(stdin, stdout, from, to, useColor, onlySameLine,
DEBUG);
} else if (numFiles == 1) { // Обработка текстового файла (вывод в консоль)
    FILE *sourceFile = fopen(argv[optind], "r");
    if (sourceFile == NULL) {
        perror("Ошибка чтения файла");
        exit(EXIT_FAILURE);
    }
    if (DEBUG) {
        fprintf(stderr, "Из файла %s:\n", argv[optind]);
    }
    processAndPrintPhoneNumbers(sourceFile, stdout, from, to, useColor,
onlySameLine, DEBUG=0);
    fclose(sourceFile);
} else if (numFiles == 2) { // Обработка текстового файла (вывод в новый файл)
    FILE *sourceFile = fopen(argv[optind], "r");
    if (useColor) {
        printf("Оно тебе нужно?\nТы же понимаешь, что выведет белиберду?\n");
    }
    if (sourceFile == NULL) {
        perror("Ошибка чтения файла");
        exit(EXIT_FAILURE);
    }
    FILE *destinationFile = fopen(argv[optind + 1], "w");
    if (destinationFile == NULL) {
        perror("Ошибка создания файла");
        fclose(sourceFile);
        exit(EXIT_FAILURE);
    }
    if (DEBUG) {
        fprintf(stderr, "\nP.S. Результат смотри в %s", argv[optind + 1]);
    }
    processAndPrintPhoneNumbers(sourceFile, destinationFile, from, to, useColor,
onlySameLine, DEBUG=0);
    fclose(sourceFile);
    fclose(destinationFile);
} else {
    fprintf(stderr, "Использование: %s [-v] [-f M] [-t N] [-c] [-n] [имя_вход_файла
[имя_выход_файла]]\n", argv[0]);
    exit(EXIT_FAILURE);
}

return EXIT_SUCCESS;
}

```

### 1.3.2 .h-файл

```
#ifndef FUNCTIONS_H
#define FUNCTIONS_H

// ANSI коды цветов
#define ANSI_COLOR_BLUE    "\x1b[34m"
#define ANSI_COLOR_RESET   "\x1b[0m"

// Ура, закончил писать код :D
// Функция обрабатывает и выводит номера телефонов из заданного входного файла в
// заданный выходной файл.
void processAndPrintPhoneNumbers(FILE *inputFile, FILE *outputFile, int from, int to,
int useColor, int onlySameLine, char *DEBUG) {
    // Регулярное выражение для поиска номеров телефонов
    const char *phoneRegex = "[+]?[78][()?[[:digit:]]{3}]?[:digit:]{3}[-
]?[:digit:]{2}[-]?[:digit:]{2}";
    regex_t regex;
    if (regcomp(&regex, phoneRegex, REG_EXTENDED) != 0) {
        fprintf(stderr, "Ошибка компиляции регулярного выражения\n");
        exit(EXIT_FAILURE);
    }

    char buffer[1024];
    int currentObject = 0;

    if (DEBUG) {
        fprintf(stderr, "Ввод: ");
    }
    while (fgets(buffer, sizeof(buffer), inputFile) != NULL) {
        if (DEBUG) {
            fprintf(stderr, "Вывод: ");
        }
        // Пропуск строк до указанного объекта "from"
        if (++currentObject < from) {
            continue;
        }
        // Поиск номеров телефонов в текущей строке
        regmatch_t match;
        char *current = buffer;
        while (regexexec(&regex, current, 1, &match, 0) == 0) {
```



```

        // Проверка, если указана опция "onlySameLine" и совпадение не находится
на той же строке
        if (onlySameLine && strchr(current, '\n') != NULL && current +
match.rm_eo > strchr(current, '\n')) {
            break; // Игнорирование совпадений, которые выходят за пределы
текущей строки
        }
        // Вывод части строки до совпадения
        fwrite(current, 1, match.rm_so, outputFile);
        // Вывод номера телефона с цветом или без него, в зависимости от опции
"useColor"
        if (useColor) {
            fprintf(outputFile, ANSI_COLOR_BLUE);
        } else {
            fputs("*", outputFile);
        }
        // Вывод части номера телефона после совпадения
        fwrite(current + match.rm_so, 1, match.rm_eo - match.rm_so, outputFile);
        // Сброс цвета, если опция "useColor" включена
        if (useColor) {
            fprintf(outputFile, ANSI_COLOR_RESET);
        } else {
            fputs("*", outputFile);
        }
        current += match.rm_eo;
    }
    fputs(current, outputFile);
    if (currentObject == to) {
        break;
    }
    if (DEBUG) {
        fprintf(stderr, "Ввод: ");
    }
}
regfree(&regex);
}

#endif

```

## **ЗАКЛЮЧЕНИЕ**

Было выполнено преобразование номеров РФ из строк [из файла] в Курсив или Синий.

Это позволило закрепить навыки работы со строками.

Я все сделал, я молодец. =)

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Лекции Гирика Алексея Валерьевича по программированию – 2023. – URL :  
<https://drive.google.com/drive/folders/1eAiMW4hD9TLhZH2vtpKPWzZkzKp10BnL>  
(дата обращения: 23.12.2023).