Московский авиационный институт

(Национальный исследовательский университет)

Факультет №3:

*Системы управления, информатика и электроэнергетика*.

Кафедра 304.

Отчёт по лабораторной работе

По учебной дисциплине «Программирование»

На тему

«Символьные данные»

Группа: М3О-107Б-23

Вариант №6

Выполнили:

*Романов Д.И.*

*Ильин А.А.*

Приняла:

*Татарникова Е.М.*

Оглавление

[Постановка задачи: 4](#_Toc152530154)

[Блок-Схема 5](#_Toc152530155)

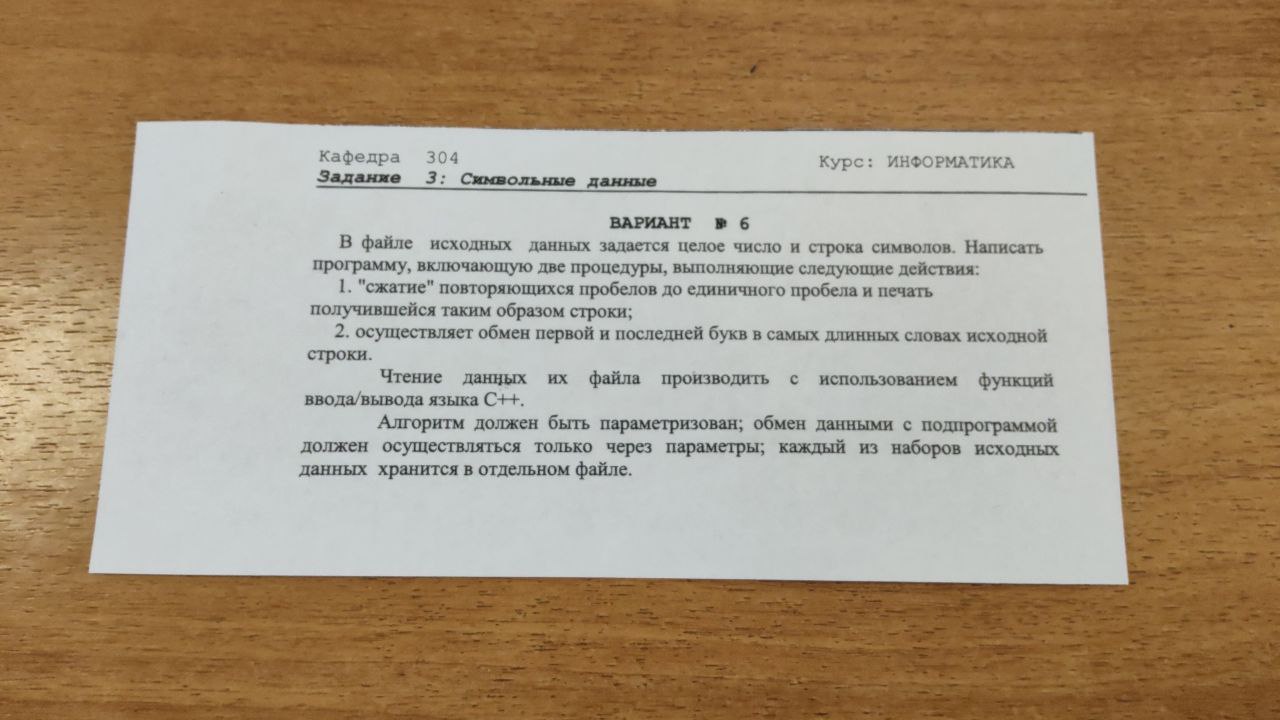
[Псевдокод 6](#_Toc152530156)

[Кодпрограммы 7](#_Toc152530157)

[Тестирование 8](#_Toc152530158)

[Вывод 9](#_Toc152530159)

# Постановка задачи:



# Блок-Схема

# 

# 

# 

# 

# Кодпрограммы

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* КАФЕДРА № 304 1 КУРС \*

\*--------------------------------------------------------------- \*

\* Project Type : Win32 Console Application \*

\* Project Name : LW1 \*

\* File Name : LW1.cpp \*

\* Language : C/C++ \*

\* Programmer(s) : Романов Д.И., Ильин А.А \*

\* Created at : 20/02/24 \*

\* Last Revision : 05/03/24 \*

\* Comment(s) : Строковые данные \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <filesystem>

#include <fstream>

const char TEST\_CASE[] = "test.txt";

const char TESTS\_FOLDER\_NAME[] = "tests";

const std::string PATH\_TO\_TEST = ((std::filesystem::current\_path() /= TESTS\_FOLDER\_NAME) /= TEST\_CASE).string();

size\_t str\_len(const char \*text); // our own ffs

void swap\_char(char &a, char &b); // same

void reset\_stream(std::ifstream &stream, const std::streampos start\_pos);

bool validate\_format(std::ifstream &in);

char \*remove\_extra\_white\_spaces(const char \*input);

char \*replace\_symbols\_in\_longest\_words(const char \*input);

int main() {

std::streamsize input\_string\_length; // Ожидаемое количество символов в строке.

char \*input\_string; // Считанный текст.

std::ifstream input\_file(PATH\_TO\_TEST); // Поток чтения из файла.

if (!input\_file.is\_open()) {

std::cerr << "Ошибка открытия файла. Убедитесь, что по указанному пути существует файл: "

<< PATH\_TO\_TEST << '\n';

return -1;

}

if (input\_file.peek() == EOF) {

std::cerr << "Файл пуст.\n";

return -2;

}

if (!validate\_format(input\_file)) {

std::cerr << "В файле должны находиться длина строки и сама строка, разделенные только пробелом. Пример:\n";

std::cerr << "13 Hello, world!\n";

return -3;

}

input\_file >> input\_string\_length;

std::cout << "Ожидаемая длина строки: " << input\_string\_length << '\n';

if (input\_string\_length < 1) {

std::cerr << "Ожидаемая длина строки не может быть меньше 1!\n";

input\_file.close();

return -4;

}

input\_string = new char[input\_string\_length + 1]; // + NULL

// we're getting only the first space

// which is the delimeter between an int and a string

input\_file.get();

// all of the following whitespace is part of the string

input\_file.getline(input\_string, input\_string\_length + 1); // + NULL

std::cout << "Полученная исходная строка: [" << input\_string << "]\n";

if (input\_file.fail()) { // there are more symbols!!1

std::cerr << "В файле больше символов, чем указано (должно быть " << input\_string\_length << ")!\n";

return -5;

}

{

size\_t real\_input\_string\_length = str\_len(input\_string);

if (real\_input\_string\_length != input\_string\_length) {

std::cerr << "В файле содержится меньше символов (" << real\_input\_string\_length << "), чем должно быть (" << input\_string\_length << ")!\n";

return -6;

}

}

if (input\_file.peek() != EOF) {

std::cerr << "В файле есть данные на строках, кроме первой!\n";

return -7;

}

const char \*text\_whithout\_extra\_white\_space = remove\_extra\_white\_spaces(input\_string);

std::cout << "Строка без лишних пробелов: [" << text\_whithout\_extra\_white\_space << "]\n";

const char\* text\_with\_replaced\_symbols = replace\_symbols\_in\_longest\_words(input\_string);

std::cout << "Текст, в котором в самых длинных словах первые и последние буквы поменялись местами: ["

<< text\_with\_replaced\_symbols << "]\n";

}

char \*remove\_extra\_white\_spaces(const char \*input) {

char \*result = new char[str\_len(input) + 1];

char \*current\_result = result;

bool last\_space = false;

while (\*input) {

if (isspace(\*input)) {

if (!last\_space) {

last\_space = true;

\*current\_result++ = \*input;

}

} else {

last\_space = false;

\*current\_result++ = \*input;

}

input++;

}

\*current\_result = '\0';

return result;

}

char \*replace\_symbols\_in\_longest\_words(const char \*input) {

char \*result = new char[str\_len(input) + 1];

char \*current\_result = result;

size\_t longest\_word = 0;

char \*word\_start = NULL;

do {

if (isspace(\*(input)) || !\*(input)) {

// we don't want it to be null, duh

if (word\_start && input - word\_start > longest\_word)

longest\_word = input - word\_start;

word\_start = NULL;

} else if (!word\_start) {

word\_start = (char \*) input;

}

\*current\_result++ = \*input++;

} while (\*input);

current\_result = result;

do {

if (isspace(\*(current\_result)) || !\*(current\_result)) {

// we don't want it to be null, duh

if (word\_start && current\_result - word\_start == longest\_word)

swap\_char(\*word\_start, \*(current\_result-1));

word\_start = NULL;

} else if (!word\_start) {

word\_start = current\_result;

}

current\_result++;

} while (\*current\_result);

std::cout << longest\_word << '\n';

return result;

}

size\_t str\_len(const char \*text) {

size\_t c = 0;

while (text[c] != '\0')

++c;

return c;

}

void swap\_char(char &a, char &b)

{

char c(a);

a = b;

b = c;

}

void reset\_stream(std::ifstream &stream, const std::streampos start\_pos) {

stream.clear();

stream.seekg(start\_pos);

}

bool validate\_format(std::ifstream &in) {

std::streamsize n;

in >> n;

// we're checking if there are actually any characters later in the code

// here we're just checking if the data is in the right format

bool c = in.peek() == ' ';

reset\_stream(in, std::ios\_base::beg);

return c;

}

# Тестирование

# Вывод

Так же был проведен ряд тестов, охватывающих область корректных и некорректных входных данных. Программа выполняет поставленную задачу, а тесты ошибки не выявили, в связи с чем было принято решение о прекращении дальнейшей разработки