|  |  |
| --- | --- |
| https://lh7-us.googleusercontent.com/QuBaagabzZYLr7U3ZbYOtMusd5FbWWxIvMUTN8jrLHzSLMg534z9gXRTIG1Us4i_lOwmWlaBxKedNt-SQ26dm4WmyqwjGDmEO6z8GE3QrZosqvHM88J2EFeVf1u0GzyCZQlhWmp1Zeo85tKo4LJVXQ | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» |

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Отчёт по лабораторной работе №1  
по учебной дисциплине «Программирование»  
на тему «Символьные данные»

Вариант №6

Выполнили  
студенты группы М3О-119БВ-24  
Шалаев А.М.,  
Нарзиев А.Т.

Приняли

ст. преподаватель каф. 304 Секретарев В.Е.,

ст. преподаватель каф. 304 Давыдкина Е.А.

Москва  
2024

**Содержание**

Задание 3

Схема алгоритма 4

Исходный код 7

Тесты программы 11

Вывод по работе 24

**Задание**

|  |  |
| --- | --- |
| Кафедра 304 | Курс: ПРОГРАММИРОВАНИЕ II семестр |
| Задание 4: *Символьные данные* | |

**ВАРИАНТ № 6**

В файле исходных данных задана строка слов. Написать программу, включающую две процедуры, выполняющие следующие действия:

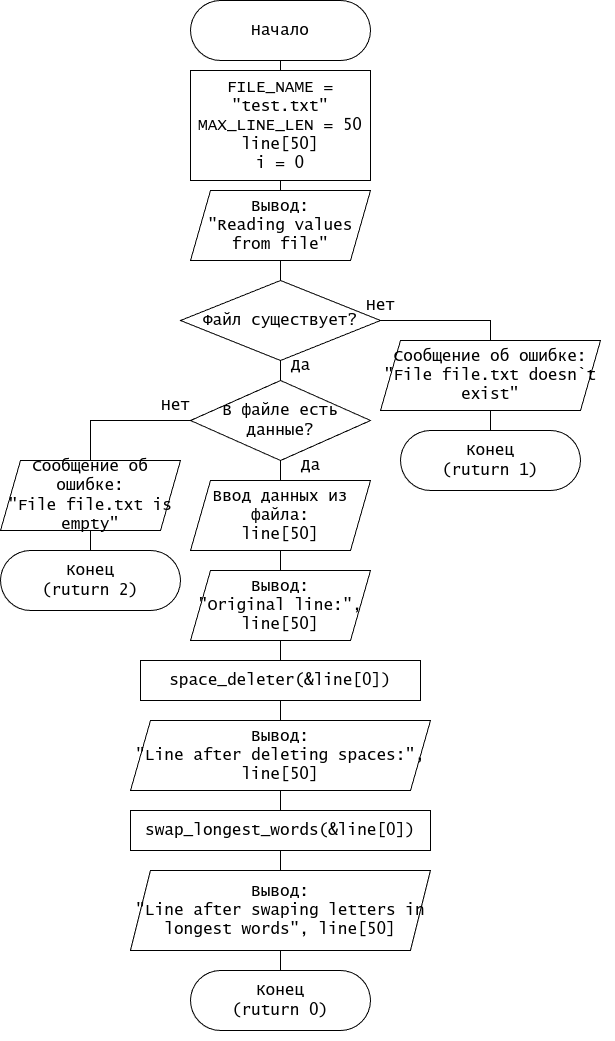
1. "сжатие" повторяющихся пробелов до единичного пробела и печать получившейся таким образом строки;

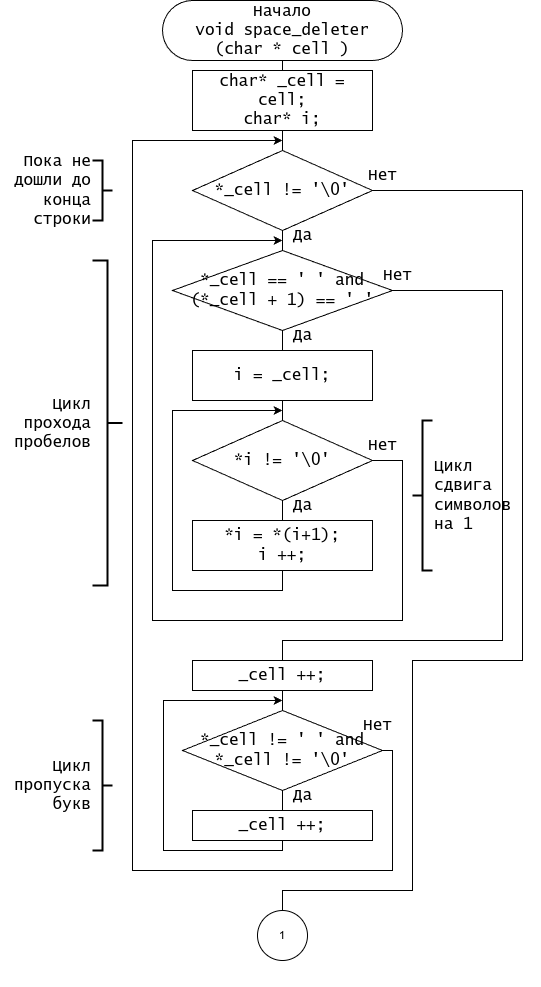
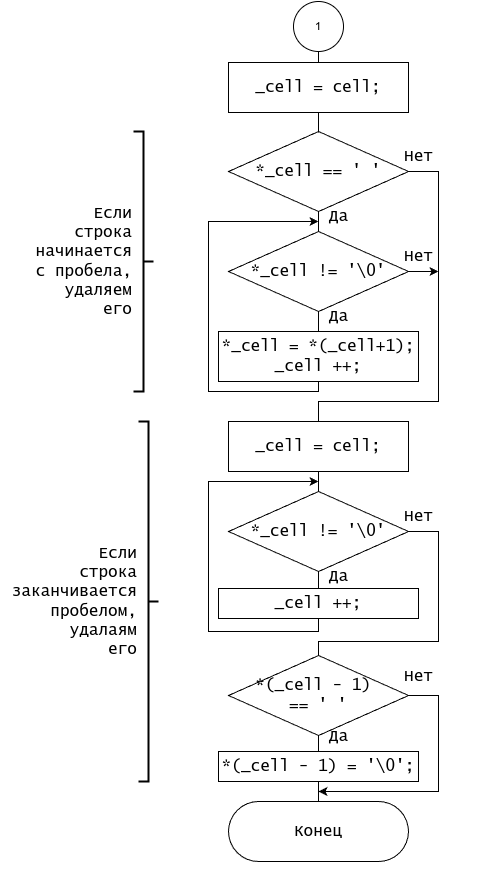
2. осуществляет обмен первой и последней букв в самых длинных словах исходной строки.

Чтение данных их файла производить с использованием функций ввода/вывода языка С++.

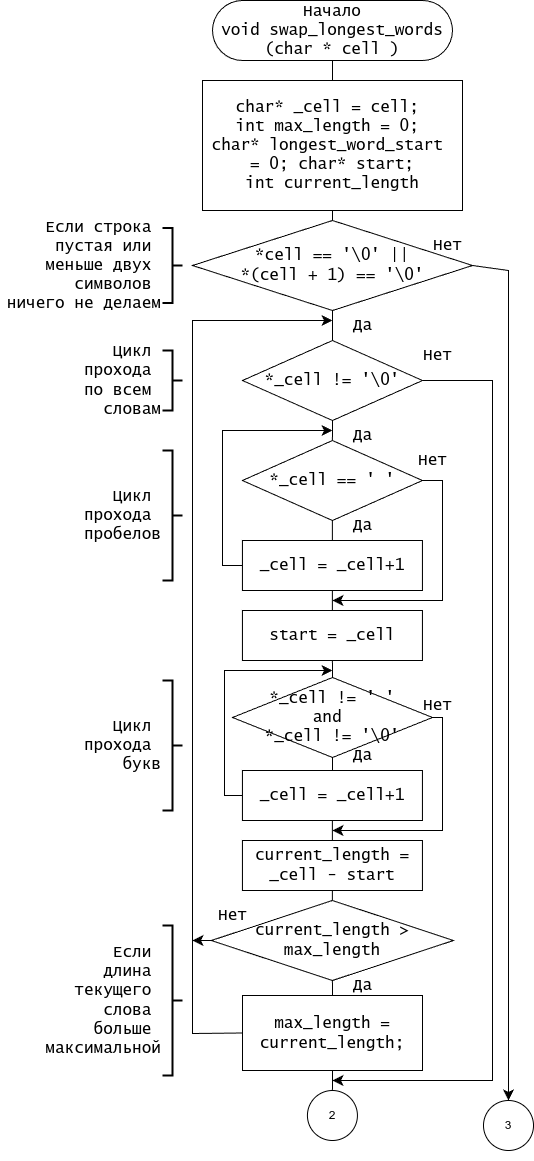
Алгоритм должен быть параметризован; обмен данными с подпрограммой должен осуществляться только через параметры; каждый из наборов исходных данных хранится в отдельном файле.

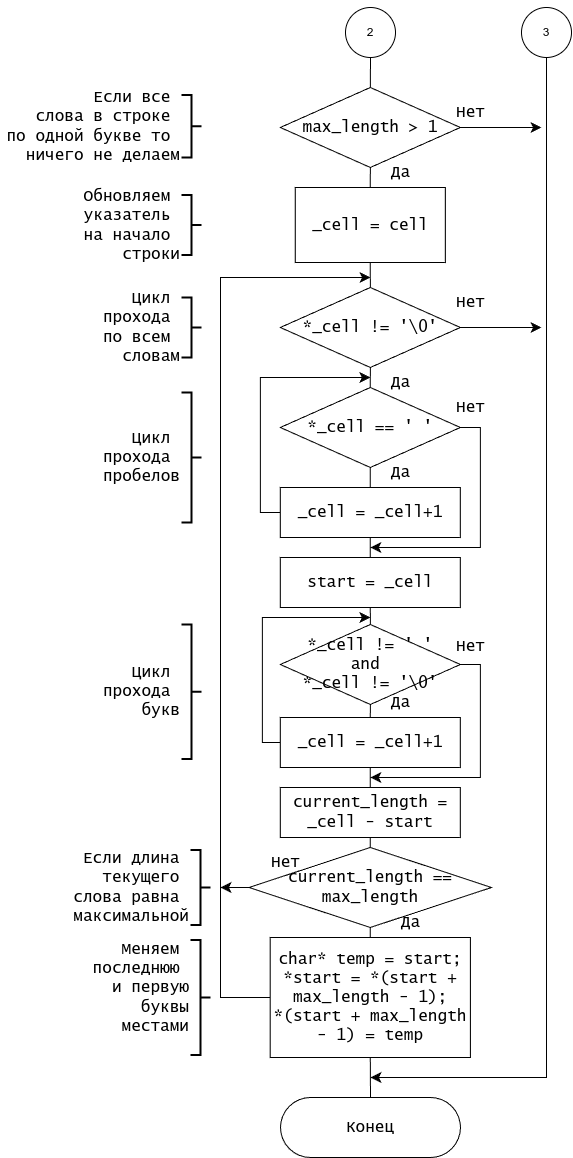
**Схема алгоритма**

main.cpp



lib.cpp – scace\_deleter





lib.cpp – swap\_longest\_words

**Исходный код**

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

\* Курс Информатика \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Project type : Linux Console Application \*

\* Project name : Lab\_1 \*

\* File name : main.cpp \*

\* Language : CPP \*

\* Programmers : Шалаев Александр Максимович, Нарзиев Артемий Тимурович \*

\* Modified By : \*

\* Created : 02.03.2025 \*

\* Last Revision : 02.03.2025 \*

\* Comment : Символьные данные. Вариант: 6 \*

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

#include "lib/lib.h"

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

const char \*FILE\_NAME = "tests/test.txt";

const int MAX\_LINE\_LEN = 100; // максимальная длина строки входных данных

int main() {

// ---- ИНИЦИАЛИЗАЦИЯ ПЕРЕМЕННЫХ ---- //

char line[MAX\_LINE\_LEN]; // строка

int i = 0; // счётчик

// ----- ЧТЕНИЕ ДАННЫХ ИЗ ФАЙЛА ----- //

cout << "Reading values from file" << FILE\_NAME << "..." << endl;

// объявление файловой переменной и открытие файла для чтения

ifstream file(FILE\_NAME);

// проверка наличия файла

if (!file.is\_open()){

cout << "File \"" << FILE\_NAME << "\" dosen`t exist\n";

file.close();

getchar(); // Пауза

return 1;

} // if

// проверка на наличие данных в файле

if (file.peek() == EOF){

cout << "File \"" << FILE\_NAME << "\" is empty\n";

file.close();

getchar();

return 2;

} // if

// запись данных в строку;

file.getline(line, MAX\_LINE\_LEN, '\n');

// -------- ВЫВОД РЕЗУЛЬТАТА -------- //

cout << setw(48) << left << "Original line"

<< "| \"" << line << "\"\n";

space\_deleter(line);

cout << setw(48) << left << "Line after deleting spaces"

<< "| \"" << line << "\"\n";

swap\_longest\_words(line);

cout << setw(48) << "Line after swapping letters in the longest words"

<< "| \"" << line << "\"\n";

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* КОНЕЦ main.cpp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

\* Курс Информатика \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Project type : Linux Console Application \*

\* Project name : Lab\_1 \*

\* File name : lib.h \*

\* Language : CPP \*

\* Programmers : Шалаев Александр Максимович, Нарзиев Артемий Тимурович \*

\* Modified By : \*

\* Created : 02.03.2025 \*

\* Last Revision : 02.03.2025 \*

\* Comment : Символьные данные. Вариант: 6 \*

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

#ifndef LIB\_H

#define LIB\_H

void space\_deleter(char\* cell);

void swap\_longest\_words(char\* cell);

#endif // LIB\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* КОНЕЦ lib.h \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

\* Курс Информатика \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Project type : Linux Console Application \*

\* Project name : Lab\_1 \*

\* File name : lib.cpp \*

\* Language : CPP \*

\* Programmers : Шалаев Александр Максимович, Нарзиев Артемий Тимурович \*

\* Modified By : \*

\* Created : 02.03.2025 \*

\* Last Revision : 02.03.2025 \*

\* Comment : Символьные данные. Вариант: 6 \*

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

#include "lib.h"

#include <iostream>

// Удаление лишних пробелов в строке.

void space\_deleter(char\* cell){

char\* \_cell = cell; // указатель на 1-ый символ строки

char\* i; // буфер для указателя

// цикл прохода по всей строке

while (\*\_cell != '\0'){

// цикл прохода пробелов

while (\*\_cell == ' ' and \*(\_cell + 1) == ' '){

i = \_cell;

// сдвиг символов на 1 назад

while (\*i != '\0'){

\*i = \*(i + 1);

i ++;

}

}

\_cell ++;

// цикл пропуска букв

while (\*\_cell != ' ' and \*\_cell != '\0'){ \_cell++;}

}

\_cell = cell;

// если строка начинается с пробела, убираем его

if (\*\_cell == ' '){

while (\*\_cell != '\0'){

\*\_cell = \*(\_cell + 1);

\_cell ++;

}

}

\_cell = cell;

// если строка заканчивается пробелом, убираем его

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

if (\*(\_cell - 1) == ' '){

\*(\_cell - 1) = '\0';

}

}

//Смена первых и последних букв в самых длинных словах строки.

void swap\_longest\_words(char \*cell){

// инициализация

char\* \_cell = cell; // Указатель по символам строки

int max\_length = 0; // Длина самого длинного слова

char\* longest\_word\_start = 0; // Указатель на начало самого длинного слова

int current\_length; // Длина текущего слова

char\* start; // Начало слова

// Eсли строка пустая или меньше 2 символов, ничего не делаем

if (\*cell == '\0' || \*(cell + 1) == '\0') {

return;

} // if

/\*

Первый проход по строке:

находим максимальную длину слова

\*/

while (\*\_cell!= '\0'){

// Цикл пропуска пробелов

while (\*\_cell == ' ') {

\_cell++;

} // while

start = \_cell; // Начало текущего слова

// Цикл прохода букв

while(\*\_cell != ' ' && \*\_cell != '\0'){

\_cell++;

} // while

current\_length = \_cell - start; // Длина текущего слова

// Обновляем максимальную длину, если текущее слово длиннее

if (current\_length > max\_length) {

max\_length = current\_length;

} // if

} // while

// Если максимальная длина слова равна 1, выходим

if (max\_length <= 1) {

return;

} // if

/\*

Второй проход по строке:

меняем первую и последнюю буквы в самых длинных словах

\*/

\_cell = cell; // Сбрасываем указатель

while (\*\_cell!= '\0'){

// Цикл пропуска пробелов

while (\*\_cell == ' ') {

\_cell++;

} // while

start = \_cell; // Начало текущего слова

// Цикл прохода букв

while(\*\_cell!= ' ' && \*\_cell!= '\0'){

\_cell++;

} // while

current\_length = \_cell - start; // Длина текущего слова

// Если слово максимальной длины, то меняем буквы

if (current\_length == max\_length) {

char temp = \*start; // Сохраняем первую букву

\*start = \*(start + max\_length - 1); // Меняем первую букву на последнюю

\*(start + max\_length - 1) = temp; // Меняем последнюю букву на первую

} // if

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* КОНЕЦ lib.cpp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

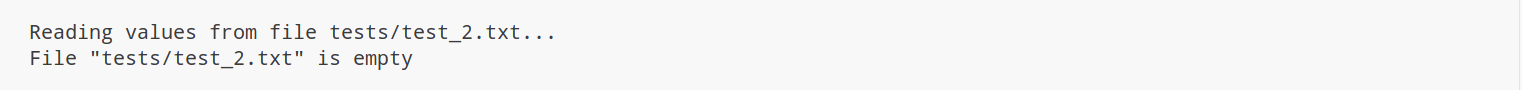
**Тесты программы**

**1) Некорректные тесты**

*Тест №1*

1. **Цель теста:** проверить работоспособность программы при отсутствии исходного файла
2. **Исходные данные:** файл test\_1.txt не существует
3. **Ожидаемый результат:**  «File "tests/test\_1.txt" dosen`t exist» (сообщение об ошибке)
4. **Полученный результат:**
5. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №2*

1. **Цель теста:** проверить работоспособность программы при пустом исходном файле
2. **Исходные данные:** файл test\_2.txt пустой
3. **Ожидаемый результат:** «File "tests/test\_2.txt" is empty»(сообщение об ошибке)
4. **Полученный результат:**
5. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.
6. **Корректные тесты**

*Тест №1*

1. **Цель теста:** Проверить работоспособность программы при корректных данных, состоящих из прдложения с разной длиной слов
2. **Исходные данные:** файл test\_3.txt с содержанием **-** “Hello world !”
3. **Ожидаемый результат:**

Line after deleting spaces | "Hello world !"

Line after swapping letters in the longest words| " oellH dorlw !"

1. **Порядок вычислений:**
   1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В результате, строка остается без изменений, так как в ней нет лишних пробелов:
   * Результат: "Hello world !"
   1. Определение самых длинных слов:

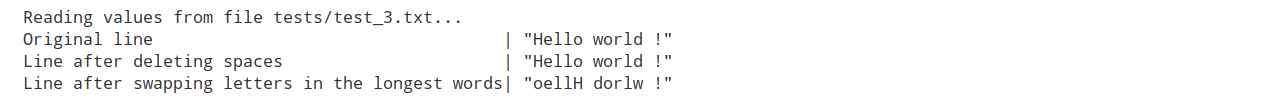
* Разбиваем строку на слова: ["Hello", "world", "!"]
* Длина слова "Hello": 5 символов.
* Длина слова "world": 5 символов.
* Длина слова "!": 1 символ.
* Самые длинные слова: "Hello" и "world" (оба имеют длину 5 символов).
  1. Замена букв в самых длинных словах:
* Для слов меняем местами первую и последнюю буквы:

Первая буква: w, последняя буква: d.

Первая буква: H, последняя буква: o.

После замены: "dorlw", "oellH"

* 1. Объединяем с остальной частью строки:
* Получаем итоговую строку: "oellH dorlw !".

1. **Полученный результат:** "oellH dorlw !".
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №2*

1. **Цель теста:** Проверить работоспособность программы при корректных данных и лишними пробелами в начале строки
2. **Исходные данные:** файл test\_4.txt с содержанием - “ I love MAI !”
3. **Ожидаемый результат:**

Line after deleting spaces | "I love MAI !"

Line after swapping letters in the longest words| "I eovl MAI !"

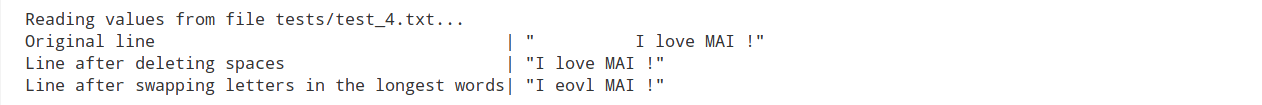
1. **Порядок вычислений:**
   1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В результате, в строке уберутся первые пробелы:
   * Результат: "I love MAI !"
   1. Определение самых длинных слов:

* Разбиваем строку на слова: ["I", "love", "MAI", "!"]
* Длина слова "I": 1 символ.
* Длина слова "love": 4 символа.
* Длина слова "MAI": 3 символа.
* Длина слова "!": 1 символ.
* Cамое длинное слово: "love" (длина 4 символа).
  1. Замена букв в самых длинных словах:
* Для слова "love" меняем местами первую и последнюю буквы:

Первая буква: l, последняя буква: e.

После замены: "eovl"

* 1. Объединяем с остальной частью строки:
* Получаем итоговую строку: "I eovl MAI !".

1. **Полученный результат:** "I eovl MAI !".
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил

*Тест №3*

1. **Цель теста:** Проверить работоспособность программы при корректных данных и лишними пробелами в середине
2. **Исходные данные:** файл test\_5.txt с содержанием - “I love MAI !”
3. **Ожидаемый результат:**

Line after deleting spaces | "I love MAI !"

Line after swapping letters in the longest words| "I eovl MAI !"

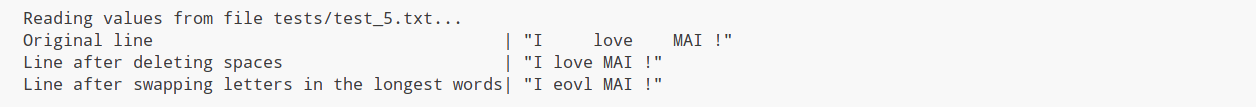
1. **Порядок вычислений:**
   1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В результате, в строке уберутся лишние пробелы между словами
   * Результат: "I love MAI !"
   1. Определение самых длинных слов:

* Разбиваем строку на слова: ["I", "love", "MAI", "!"]
* Длина слова "I": 1 символ.
* Длина слова "love": 4 символа.
* Длина слова "MAI": 3 символа.
* Длина слова "!": 1 символ.
* Cамое длинное слово: "love" (длина 4 символа).
  1. Замена букв в самых длинных словах:
* Для слова "love" меняем местами первую и последнюю буквы:

Первая буква: l, последняя буква: e.

После замены: "eovl"

* 1. Объединяем с остальной частью строки:
* Получаем итоговую строку: "I eovl MAI !".

1. **Полученный результат:** "I eovl MAI !"
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №4*

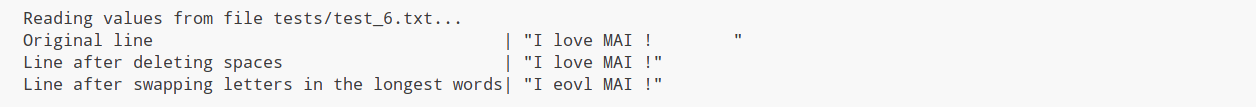
1. **Цель теста:**Проверить работоспособность программы при корректных данных и лишними пробелами в конце
2. **Исходные данные:** файл test\_6.txt с содержанием - “I love MAI ! ”
3. **Ожидаемый результат:**

Line after deleting spaces | "I love MAI !"

Line after swapping letters in the longest words| "I eovl MAI !"

1. **Порядок вычислений:**
   * 1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В результате, строка остается без изменений, так как в ней нет лишних пробелов.
   * Результат: "I love MAI !"
     1. Определение самых длинных слов:
   * Разбиваем строку на слова: ["I", "love", "MAI", "!"]
   * Длина слова "I": 1 символ.
   * Длина слова "love": 4 символа.
   * Длина слова "MAI": 3 символа.
   * Длина слова "!": 1 символ.
   * Cамое длинное слово: "love" (длина 4 символа).
     1. Замена букв в самых длинных словах:
   * Для слова "love" меняем местами первую и последнюю буквы
   * Первая буква: l, последняя буква: e.
   * После замены: "eovl"
     1. Объединяем с остальной частью строки:

* Получаем итоговую строку: "I eovl MAI !".

1. **Полученный результат:** "I eovl MAI !"
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №5*

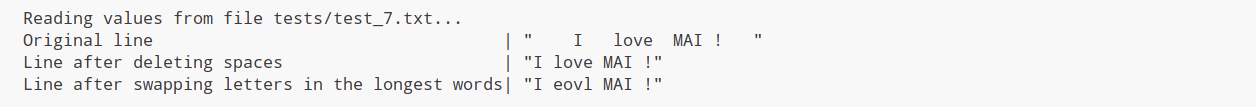
1. **Цель теста:** Проверить работоспособность программы при корректных данных и лишними пробелами во всей строке
2. **Исходные данные:** файл test\_7.txt с содержанием - “ I love MAI ! ”
3. **Ожидаемый результат:**

Line after deleting spaces | "I love MAI !"

Line after swapping letters in the longest words| "I eovl MAI !"

1. **Порядок вычислений:**
   * 1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В результате уберутся лишние пробелы в начале и конце строки
   * Результат: "I love MAI !"
     1. Определение самых длинных слов:
   * Разбиваем строку на слова: ["I", "love", "MAI", "!"]
   * Длина слова "I": 1 символ.
   * Длина слова "love": 4 символа.
   * Длина слова "MAI": 3 символа.
   * Длина слова "!": 1 символ.
   * Cамое длинное слово: "love" (длина 4 символа).
     1. Замена букв в самых длинных словах:
   * Для слова "love" меняем местами первую и последнюю буквы
   * Первая буква: l, последняя буква: e.
   * После замены: "eovl"
     1. Объединяем с остальной частью строки:

* Получаем итоговую строку: "I eovl MAI !".

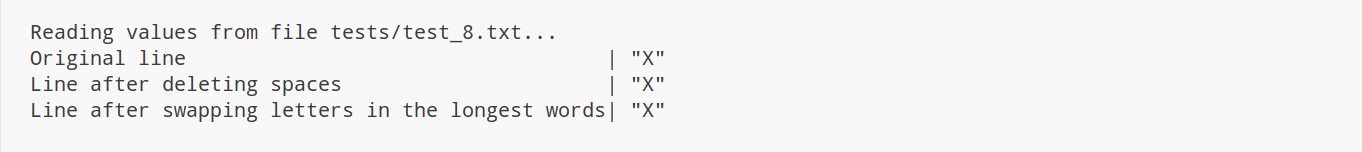
1. **Полученный результат:** "I eovl MAI !"
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №6*

1. **Цель теста:** Проверить работоспособность программы при корректных данных, состоящих из одной буквы
2. **Исходные данные:** файл test\_8.txt с содержанием - “X”
3. **Ожидаемый результат:**

Line after deleting spaces | "X"

Line after swapping letters in the longest words| "X"

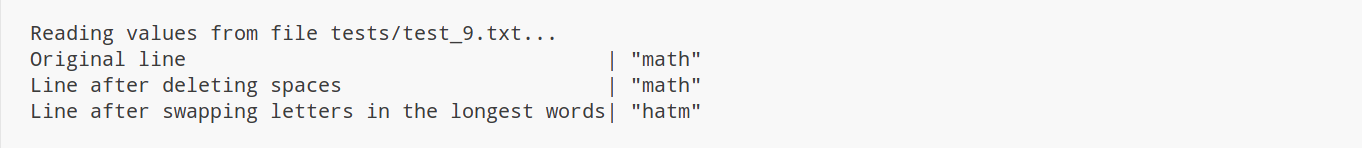
1. **Порядок вычислений:**
   * 1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В данной строке нет пробелов, поэтому результат остается прежним.
   * Результат: "X"
     1. Определение самых длинных слов:
   * Разбиваем строку на слова: ["X"]
   * Cамое длинное длинное слово: "X" (длина 1 символ).
     1. Замена букв в самых длинных словах:
   * Для слова "X" нет возможности поменять местами буквы, так как в слове только одна буква.
   * Итоговая строка остается без изменений: "X".
2. **Полученный результат:**
3. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №7*

1. **Цель теста:** Проверить работоспособность программы при корректных данных, состоящих из одного слова
2. **Исходные данные:** файл test\_9.txt с содержанием - “math”
3. **Ожидаемый результат:**

Line after deleting spaces | "math"

Line after swapping letters in the longest words| "hatm"

1. **Порядок вычислений:** 
   * 1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В данной строке нет пробелов, поэтому результат остается прежним
   * Результат: " math"
     1. Определение самых длинных слов:
   * Разбиваем строку на слова: ["math"]
   * Cамое длинное слово: "math" (длина 4 символа).
     1. Замена букв в самых длинных словах:
   * Для слова " math" меняем местами первую и последнюю буквы
   * Первая буква: m, последняя буква: h.
   * После замены: "hatm"
2. **Полученный результат:**
3. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №8*

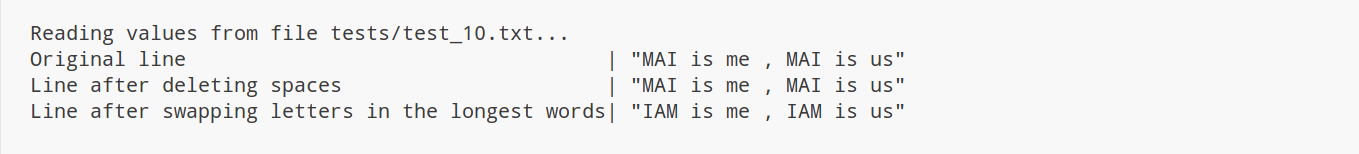
1. **Цель теста:** Проверить работоспособность программы при корректных данных, состоящих предложения с двумя словами одинаковой длины
2. **Исходные данные:** файл test\_10.txt с содержанием - “MAI is me , MAI is us”
3. **Ожидаемый результат:**

Line after deleting spaces | "MAI is me , MAI is us"

Line after swapping letters in the longest words| "IAM is me , IAM is us"

1. **Порядок вычислений:**
   * 1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В данной строке между словами нет лишних пробелов, поэтому результат остается прежним.
   * Результат: "MAI is me , MAI is us"
     1. Определение самых длинных слов:
   * Разбиваем строку на слова: ["MAI", "is", "me", ",", "MAI", "is", "us"]
   * Длина слова "MAI": 3 символа.
   * Длина слова "is": 2 символа.
   * Длина слова "me": 2 символа.
   * Длина слова ",": 1 символ.
   * Длина слова "us": 2 символа.
   * Cамые длинные слова: "MAI" (длина 3 символа). Поскольку "MAI" встречается дважды, меняем в каждом из них.
     1. Замена букв в самых длинных словах:
   * Для слова "MAI" меняем местами первую и последнюю буквы
   * Первая буква: M, последняя буква: I.
   * После замены: "eovl"
     1. Объединяем с остальной частью строки:

* Получаем итоговую строку: "IAM is me , IAM is us".

1. **Полученный результат:** "IAM is me , IAM is us".
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №9*

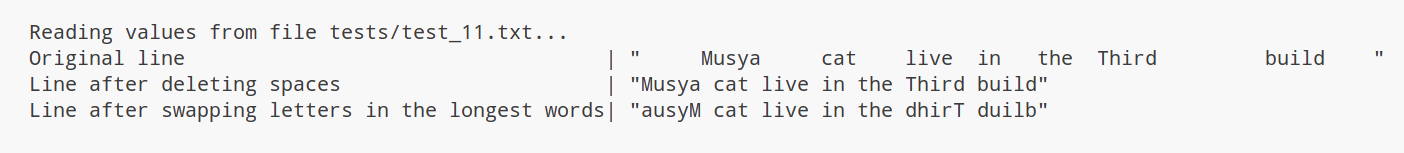
1. **Цель теста:** Проверить работоспособность программы при корректных данных, состоящих предложения с несколькими словами одинаковой длины и лишними пробелами по всей длине строки
2. **Исходные данные:** файл test\_11.txt с содержанием - “ Musya cat live in the Third build ”
3. **Ожидаемый результат:**

Line after deleting spaces | "Musya cat live in the Third build"

Line after swapping letters in the longest words| "ausyM cat live in the dhirT duilb"

1. **Порядок вычислений:**
   * 1. Удаление пробелов
   * Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
   * В данной строке между словами нет лишних пробелов, поэтому результат остается прежним.
   * Результат: "Musya cat live in the Third build"
     1. Определение самых длинных слов:
   * Разбиваем строку на слова: ["Musya", "cat", "live", "in", "the", "Third", "build"]
   * Длина слова "Musya": 5 символов.
   * Длина слова "cat": 3 символа.
   * Длина слова "live": 4 символа.
   * Длина слова "in": 2 символа.
   * Длина слова "the": 3 символа.
   * Длина слова "Third": 5 символов.
   * Длина слова "build": 5 символов.
   * Cамые длинные слова: "Musya", "Third" и "build" (длина 5 символов).
     1. Замена букв в самых длинных словах:
   * Для слова "Musya" меняем местами первую и последнюю буквы
   * Первая буква: M, последняя буква: a.
   * После замены: "ausyM"
   * Для слова "Third" меняем местами первую и последнюю буквы
   * Первая буква: T, последняя буква: d.
   * После замены: "dhirT"
   * Для слова "build" меняем местами первую и последнюю буквы
   * Первая буква: b, последняя буква: d.
   * После замены: "duilb"
     1. Объединяем с остальной частью строки:

* Получаем итоговую строку: "ausyM cat live in the dhirT duilb".

1. **Полученный результат:** "ausyM cat live in the dhirT duilb".
2. **Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

*Тест №10*

1. **Цель теста:** Проверить работоспособность программы при корректных данных, состоящих исключительно из пробелов
2. **Исходные данные:** файл test\_12.txt с содержанием - “ ”
3. **Ожидаемый результат:**
4. **Порядок вычислений:**
5. Удаление пробелов

* Функция space\_deleter проходит по строке и удаляет лишние пробелы, оставляя только один пробел между словами и убирая пробелы в начале и конце строки.
* В данной строке нет слов, только пробелы, поэтому после удаления пробелов результат будет пустой строкой.
* Результат: ""

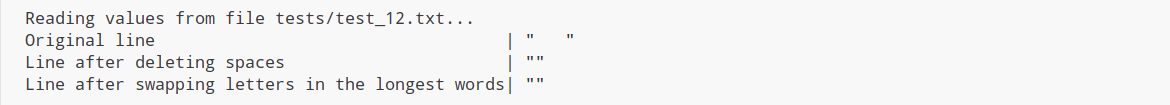
1. Определение самых длинных слов:

* Разбиваем строку на слова: [] (пустой массив, так как строка пустая).
* Так как нет слов, нет и длинных слов.
* Самые длинные слова: [] (пустой массив).

1. Замена букв в самых длинных словах:

* Так как нет самых длинных слов, нет и замены букв.
* Результат: ""

1. **Полученный результат:** ""

**Вывод:** Полученный результат совпал с ожидаемым, тест ошибку не обнаружил.

**Вывод по работе**

Разработка программы завершена на том основании, что:

1. Полученные результаты совпали с ожидаемыми
2. Считаем набор тестов полным