**Московский авиационный институт**

**(национальный исследовательский институт)**

Институт №3

Системы управления, информатики и электроэнергетики

Кафедра 304

Вычислительный машины, системы и сети

**Отчет по лабораторной работе**

**по учебной дисциплине**

**«Компьютерные технологии»**

**на тему:**

***«Символьные данные»***

Группа: М30-109Б-19

Вариант: №9

Выполнили:

*Кравченко Д.С.., Кузнецов И.И.*

Приняли:

*Давыдкина Е.А.,*

*Секретарев В.Е.*

*Москва 2019*

**Содержание.**

**Задание**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

**Блок-схема**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4-5

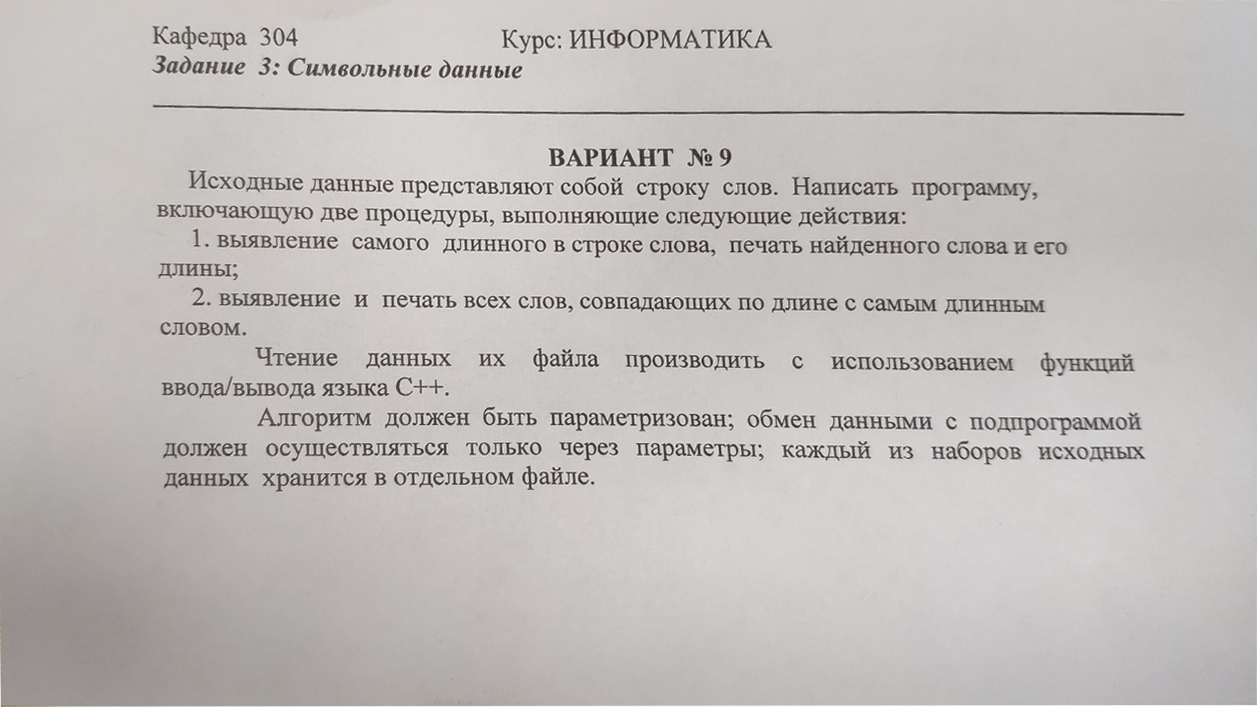
**Псевдокод**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6-7

**Код программы**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8-9

**Тесты**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10-14

**Вывод**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15

**Задание**







**Псевдокод:**

**АЛГ**

**НАЧАЛО**

*Создать объект класса fin*

*Открытие файла*

**ЕСЛИ** ФАЙЛ НЕ НАЙДЕН

**ТО**

*Вывести: Ошибка! Файл не найден!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

**ЕСЛИ** НАЙДЕННЫЙ ФАЙЛ ПУСТ

**ТО**

*Вывести: Ошибка! Файл пуст!*

*Закрыть программу*

**КОНЕЦ ЕСЛИ**

Строковые переменные:

str

*СЧИТАТЬ СТРОКУ*

*Закрыть файл*

*Вывести: Изначальная строка: str*

Натуральные аргументы*:*

i = 0

IndexFirstSymbol

IndexLastSymbol

MaxLenghtWord = 0

**НАЧАЛО ЦИКЛА ПОКА** НЕ КОНЕЦ СТРОКИ

**НАЧАЛО ЦИКЛА ПОКА** НЕ КОНЕЦ СТРОКИ **И** ОЧЕРЕДНОЙ СИМВОЛ НЕ ПРОБЕЛ

Увеличиваем счетчик i на 1

**КОНЕЦ ЦИКЛА**

IndexFirstSymbol = i

**НАЧАЛО ЦИКЛА ПОКА** НЕ КОНЕЦ СТРОКИ **И** ОЧЕРЕДНОЙ СИМВОЛ НЕ ПРОБЕЛ

Увеличиваем счетчик i на 1

**КОНЕЦ ЦИКЛА**

IndexLastSymbol = i;

**НАЧАЛО ЕСЛИ** (IndexLastSymbol - IndexFirstSymbol) > MaxLenghtWord)

MaxLenghtWord = IndexLastSymbol - IndexFirstSymbol;

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

*Вывести: Максимальная длина слова: MaxLenghtWord*

*i=0 // возвращаемся на начало строки*

*Вывести:* *Слова с длиной MaxLenghtWord:*

**НАЧАЛО ЦИКЛА ПОКА** НЕ КОНЕЦ СТРОКИ

**НАЧАЛО ЦИКЛА ПОКА** НЕ КОНЕЦ СТРОКИ **И** ОЧЕРЕДНОЙ СИМВОЛ НЕ ПРОБЕЛ

Увеличиваем счетчик i на 1

**КОНЕЦ ЦИКЛА**

IndexFirstSymbol = i

**НАЧАЛО ЦИКЛА ПОКА** НЕ КОНЕЦ СТРОКИ **И** ОЧЕРЕДНОЙ СИМВОЛ НЕ ПРОБЕЛ

Увеличиваем счетчик i на 1

**КОНЕЦ ЦИКЛА**

IndexLastSymbol = i;

**НАЧАЛО ЕСЛИ** (IndexLastSymbol - IndexFirstSymbol) == MaxLenghtWord)

**НАЧАЛО ЦИКЛА** ПО i сIndexFirstSymbol до IndexLastSymbol

*Вывести:* *очередной символ*

**КОНЕЦ ЦИКЛА**

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**КОНЕЦ**

**Код программы:**

#include <string>

#include <fstream>

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

ifstream fin;

fin.open("Text.txt");

if (!fin) // если файл не найден

{

cout << "Ошибка! файл не найден" << endl; //вывод сообщения об ошибки

fin.close(); //закрыть файл

system("pause");

return 1;

}

if (fin.peek() == EOF) //если найденный файл пуст

{

cout << "Ошибка! файл пуст!" << endl;// вывод сообщения об ошибке

fin.close();//закрыть файл

system("pause");

return 1;

}

string str;

getline(fin, str);

fin.close();

cout << "Изначальная строка: " << str << endl;

int i = 0;

int IndexFirstSymbol;

int IndexLastSymbol;

int MaxLenghtWord = 0;

//поиск самого длинного слова

while (str[i]) {

//найти первый символ слова

while ((str[i] == ' ') && (str[i] != '\0')) {

i++;

}

IndexFirstSymbol = i;

//найти последний символ слова

while ((str[i] != ' ') && (str[i] != '\0')) {

i++;

}

IndexLastSymbol = i;

//найти максимальную длину слова

if ((IndexLastSymbol - IndexFirstSymbol) > MaxLenghtWord) {

MaxLenghtWord = IndexLastSymbol - IndexFirstSymbol;

}

}

cout << "Максимальная длина слова: " << MaxLenghtWord << endl;

i = 0; //возвращаемся на начало строчки

cout << "Слова с длиной '" << MaxLenghtWord << "':" << endl;

//поиск слов с аналогичной длиной

while (str[i]) {

//найти первый символ слова

while ((str[i] == ' ') && (str[i] != '\0')) {

i++;

}

IndexFirstSymbol = i;

//найти последний символ слова

while ((str[i] != ' ') && (str[i] != '\0')) {

i++;

}

IndexLastSymbol = i;

//проверка на длину слова

if (IndexLastSymbol - IndexFirstSymbol == MaxLenghtWord) {

for (int i = IndexFirstSymbol; i < IndexLastSymbol; i++) {

cout << str[i];

}

cout << " ";

}

}

cout << endl;

return 0;

}

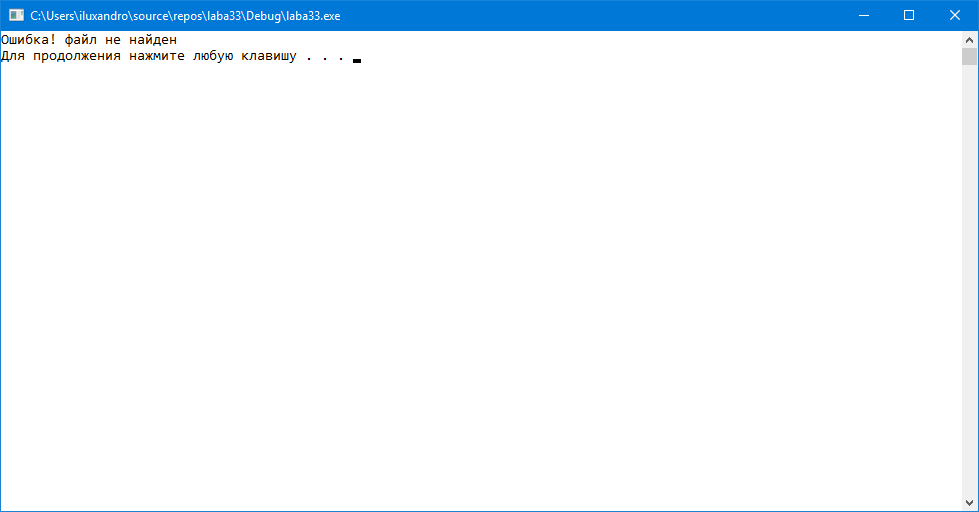
**Тесты**

**Тест №1.**

**Цель:** проверить работу программы с отсутствующим файлом

**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



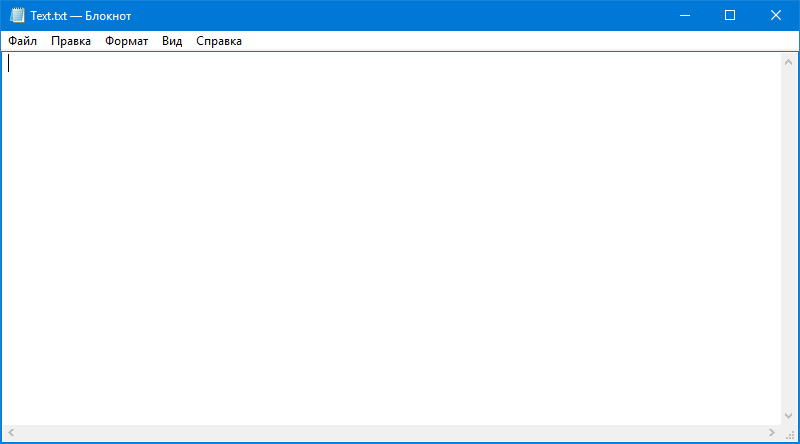
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №2.**

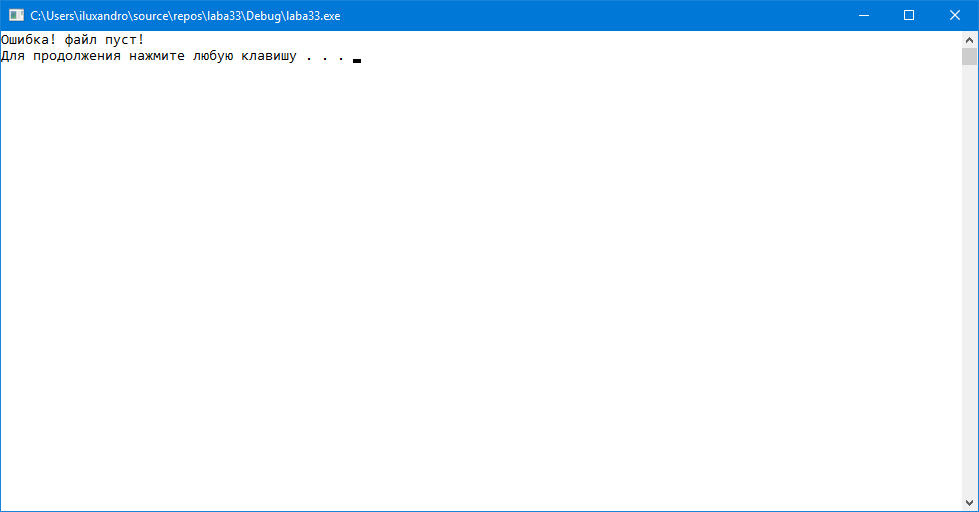
**Цель:** проверить работу программы с пустым файлом

**Ввод:**

****

**Ожидаемые результаты:** Вывод сообщения об ошибке

**Полученные результаты:**



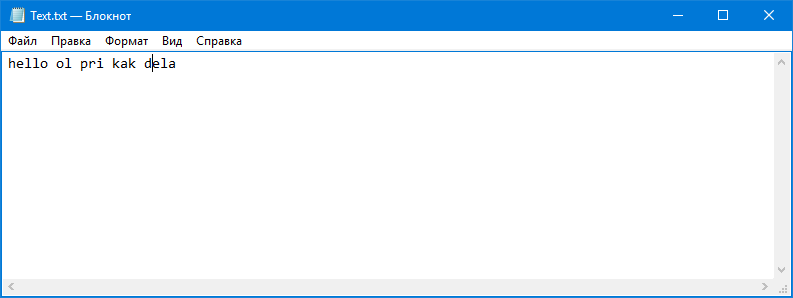
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №3.**

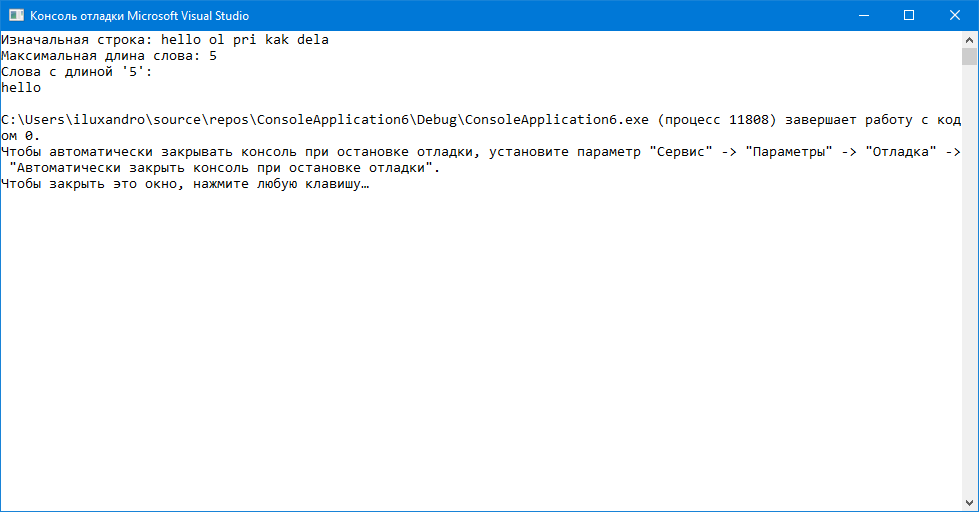
**Цель:** проверить работу программы

**Ввод:**



**Ожидаемые результаты:** Корректная работа программы

**Полученные результаты:**



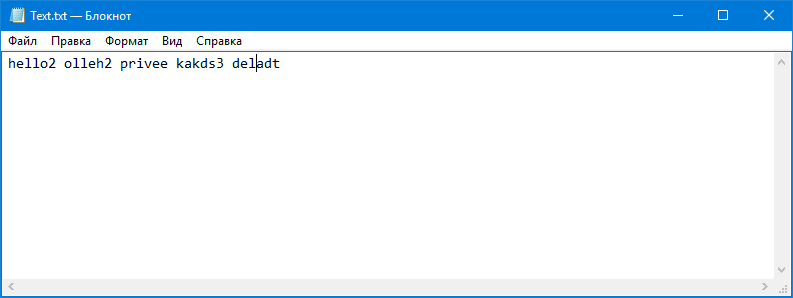
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №4.**

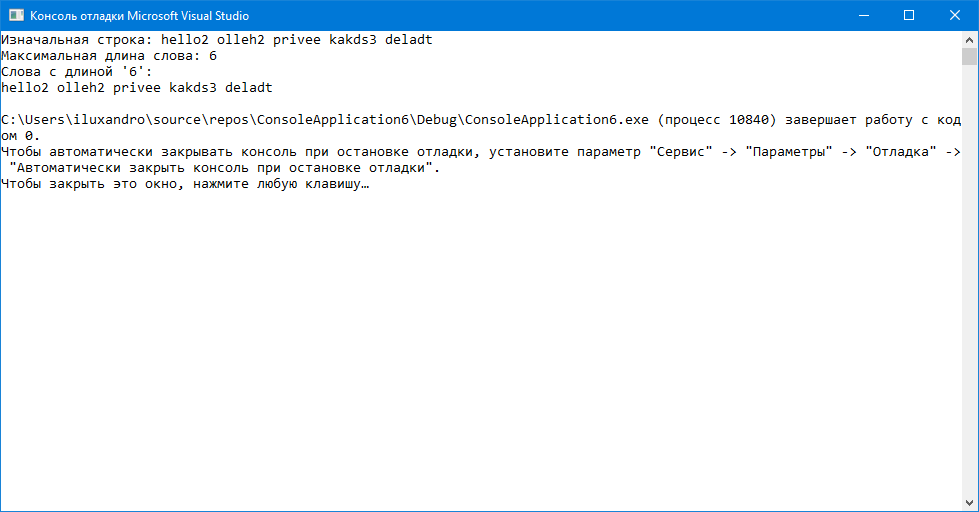
**Цель:** проверить работу программы со словами одинаковой длины

**Ввод:**



**Ожидаемые результаты:** Корректная работа программы

**Полученные результаты:**



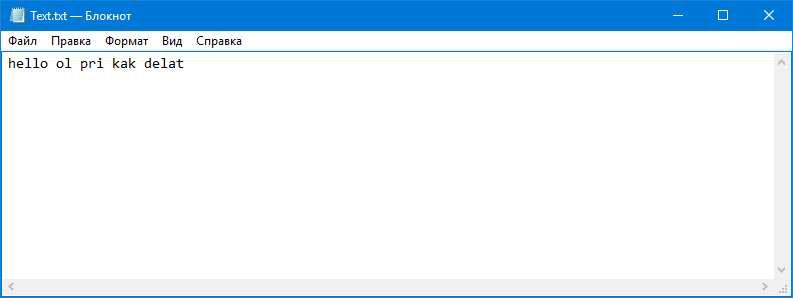
**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Тест №5.**

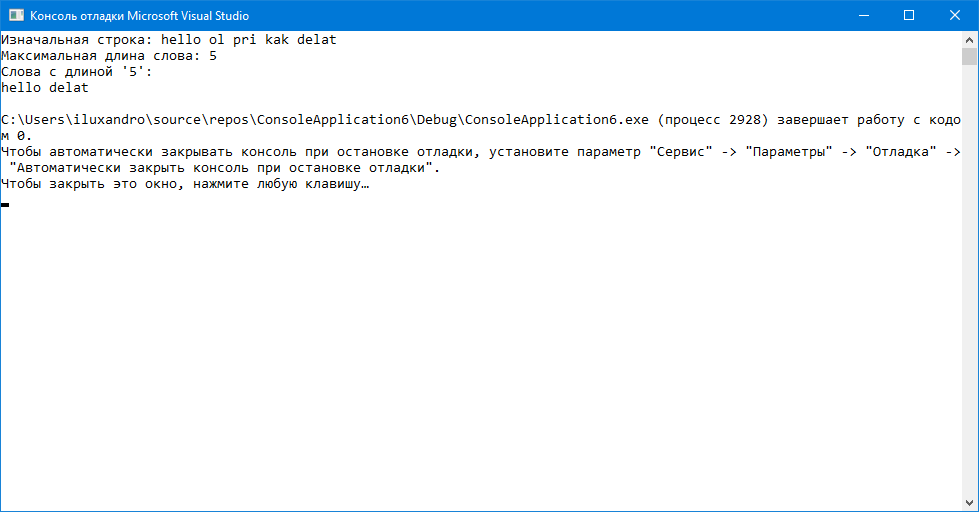
**Цель:** проверить работу программы с двумя словами одинаковой длины

**Ввод:**



**Ожидаемые результаты:** Корректная работа программы

**Полученные результаты:**



**Вывод**: Полученные результаты равны ожидаемым.

Тест не выявил ошибок. Работа программы корректна

**Вывод:**

Разработка программы успешно завершена, так как:

**1)**Полученные результаты совпадают с ожидаемыми.

**2)**Набор тестов считаем полным.