МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №12

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Строки и текстовые файлы»

Работу выполнил

студент гр. 4941 Н.С. Горбунов

Санкт-Петербург

2020

**Цель лабораторной работы:** изучение структурной организации, способов доступа к элементам и других особенностей текстовых файлов; изучение стандартных средств языка C/C++ для работы с символами, строками и текстовыми файлами; совершенствование навыков процедурного программирования на языке C/C++ при решении задач редактирования текстовых файлов.

**Задание на программирование:** используя технологию процедурного программирования разработать программу обработки текстовых файлов с числом строк не менее пяти, каждая из которых содержит не более 80 символов, в соответствии с индивидуальным заданием.

**Вариант 13**

Дан текст, составленный из строк. Под *словом* текста понимается последовательность букв алфавита. Между соседними словами - не менее одного пробела. Перед первым и за последним словом каждой строки произвольное число пробелов.

Удалить в каждом слове его последнюю букву.

**Математическая модель**Для решения задачи потребуется реализовать несколько функций, таких как:

1. Функция проверки наличия файла директории.  
    Если файл с таким именем существует, то возвращает TRUE, иначе FALSE.
2. Функция создания нового файла MakeFile.

В начале производится проверка на наличие файла с таким именем, если есть, то пользователю предлагается задать новое имя файла, или перезаписать.

После этого открывается файл на запись, и пользователь построчно вводит текст в файл. Пустая строка означает конец ввода.

В конце файл закрывается, и функция возвращает по ссылке имя созданного файла.

1. Функция вывода содержимого файла OutputFile.

Функция построчно читает и выводит в консоль текст из файла.

1. Функция удаления последнего символа каждого слова в файле DelLastSymb

Открывается два файла: один на чтение, другой на запись. Из файла построчно читается текст, разделяется на слова, и в каждом слове последний символ заменяется на “\0”. Если в слове одна буква, то она не удаляется, чтобы не потерять слово. Слова обрабатываются поочередно, записываются в строку. Строка записывается файл, оба файла закрываются.

**Текст программы**

/\*Программа создает файл строк. Признак окончания ввода - пустая строка.

Затем файл построчно читается, и программа удаляет последнюю букву

Обработанная строка записывается в новый файл\*/

#include<stdio.h>

#include<iostream>

#include<string>

#include<stdlib.h>

#include<locale.h>

#include<windows.h>

using namespace std;

#define FNAME\_I "D:\\st\_input.txt\0"//имя исх. файла задано константной строкой

#define FNAME\_R "D:\\st\_output.txt\0"//имя рез. файла задано константной строкой

const int RAZ = 81; //максимальная длина строки

const char PR\_R[] = "r"; //признак открытия файла на чтение

const char PR\_W[] = "w"; //признак открытия файла на запись

void MakeFile(char\*); //функция создания файла

bool CheckFile(char\*, const char\*); //функция проверки наличия файла

void DelLastSymb(char\*, char\*); //функция обработки содержимого

void OutputFile(char\*); //вывод содержимого файла на печать

int main()

{

char fname\_i[] = FNAME\_I;

char fname\_r[] = FNAME\_R;

setlocale(LC\_ALL, "RUSSIAN"); //Чтобы русский текст поддерживался

SetConsoleCP(1251); //меняем кодировку консоли принудительно

SetConsoleOutputCP(1251); //меняем кодировку консоли принудительно на вывод

cout << " Программа создает файл строк." << endl <<

" Затем файл построчно читается, при этом в каждой строке," << endl <<

" содержащей слова, разделенные произвольным количеством пробелов," << endl << " последние буквы строк удаляются" << endl <<

" Создаем исходный файл:" << endl;

//создаем исходный файл

MakeFile(fname\_i);

//выводим его содержимое

if (!CheckFile(fname\_i, PR\_R))

{

cout << " Ошибка открытия файла " << fname\_i << " на чтение" << endl << " Нажмите <Enter>" << endl;

getchar();

return 0; //если исходный файл не создан

}

cout << " Содержимое исходного файла:" << endl;

OutputFile(fname\_i);

//обрабатываем файл

DelLastSymb(fname\_i, fname\_r);

//выводим содержимое результата

if (!CheckFile(fname\_r, PR\_R))

{

cout << " Ошибка открытия файла " << fname\_r << " на чтение" << endl <<

" Нажмите <Enter>" << endl;

getchar();

return 0; //если результирующий файл не создан

}

cout << " Результат (удалены последние буквы слов):" << endl;

OutputFile(fname\_r);

cout << endl << " Для завершения нажмите <Enter>";

getchar();

return 0;

}

//функция создания файла

void MakeFile(char\* fname\_i)

{

char st[RAZ]; //исходная строка

FILE\* in; //текстовый файл

char otv; //ответ пользователя

/\*Открываем файл в режиме чтения (r)

Если файл с таким именем уже есть, то либо пользователь вводит

новое имя, либо новые записи будут записаны поверх старых\*/

while (CheckFile(fname\_i, PR\_R))

{

cout << " Файл с заданным именем уже есть! Зададите новое имя Y/N? ";

cin.get(otv);

if (otv == 'Y' || otv == 'y' || otv == 'Н' || otv == 'н')

{

cout << " Введите новое имя:" << endl;

cin >> fname\_i;

cin.get(); //убираем символ '\n' из потока

}

else {

cin.get(); //убираем символ '\n' из потока

break;

}

}

/\*проверяем возможность открытия файла для записи

поскольку полное имя файла могло быть задано с ошибкой\*/

if (!CheckFile(fname\_i, PR\_W))

{

cout << " Ошибка открытия файла " << fname\_i << " на запись" << endl

<< " Нажмите <Enter>" << endl;

getchar();

exit(0);

}

/\*Открываем файл в режиме записи (w) текста (по умолчанию).

Если файл с таким именем уже есть, то новые данные

будут записаны поверх старых\*/

in = fopen(fname\_i, PR\_W);

cout << " Создание файла." << endl <<

" Признак окончания ввода - ввод пустой строки" << endl <<

" Введите строку и нажмите <Enter>" << endl << " ->";

cin.getline(st, RAZ); /\*функция вводит строку целиком, включая

пробелы и символ \n\*/

while (strlen(st))

{

fprintf(in, "%s\n", st);

cout << " Введите строку и нажмите <Enter>" << endl << " ->";

cin.getline(st, RAZ);

}

fclose(in); //закрываем файл

}

//функция проверки наличия файла

bool CheckFile(char\* fname, const char\* pr)

{

//Открываем файл

if ((fopen(fname, pr)) == NULL) //если файл не существует

return(false);

return(true);

}

//функция обработки содержимого файла

void DelLastSymb(char\* fname\_i, char\* fname\_r)

{

FILE\* in, \* out; //исходный и результирующий файлы (потоки)

char st\_in[RAZ]; //обрабатываемая строка

char st\_out[RAZ]; //результирующая строка

char\* p = new char(RAZ); //указатель на выделенное слово

in = fopen(fname\_i, PR\_R); //открываем файл на чтение

out = fopen(fname\_r, PR\_W); //открываем файл на запись

fgets(st\_in, RAZ, in); //читаем первую строку из файла in

while (!feof(in)) //пока не найден признак конца файла

{

st\_out[0] = '\0'; //очищаем результирующую строку

p = strtok(st\_in, " \n"); //выделяем первое слово (лексему).

while (p) //Пока очередное слово выделено

{//обрабатываем слово

if (strlen(p) > 1) //если длина слова больше 1

{

if (ispunct(p[strlen(p) - 1])) //если знак пунктуации

{

p[strlen(p) - 2] = p[strlen(p) - 1]; // то знак скопировать на место последней буквы

p[strlen(p) - 1] = '\0'; // и удалить его на старом месте

}

else

{

p[strlen(p) - 1] = '\0'; // либо удалить последнюю букву

}

}

strcat(st\_out, p); //добавляем слово к результату без последней буквы

strcat(st\_out, " "); //добавляем пробел между слов

p = strtok(NULL, " \n"); //выделение последующих слов

}//while(p)

st\_out[strlen(st\_out) - 1] = '\0'; //убираем последний пробел

if (strlen(st\_out)) //если строка не пустая

fprintf(out, "%s\n", st\_out); //записываем строку в новый файл

fgets(st\_in, RAZ, in); //считываем новую строку из файла

}//while(!feof(in))

fclose(in); //Закрываем файл in

fclose(out); //Закрываем файл out

}

//функция вывода содержимого файла на экран

void OutputFile(char\* fname)

{

FILE\* in\_out;

char st[RAZ]; //считанная строка

in\_out = fopen(fname, PR\_R); //открываем файл на чтение

fgets(st, RAZ, in\_out); //читаем строку из файла in\_out

while (!feof(in\_out)) //пока не найден признак конца файла

{

cout << st; //выводим очередную строку на экран

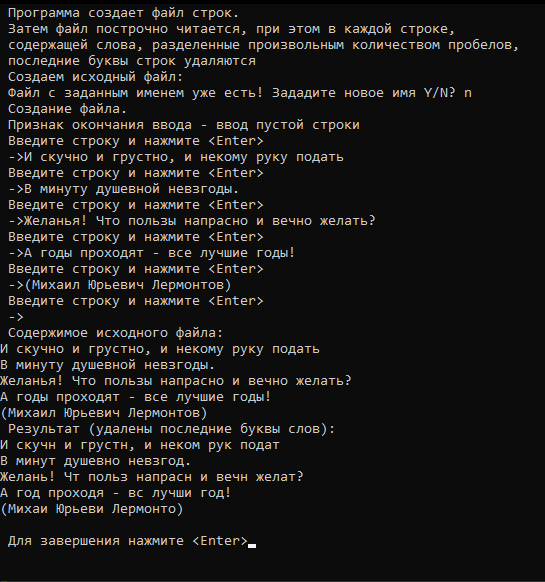
fgets(st, RAZ, in\_out); //читаем строку из файла in\_out

}

fclose(in\_out); //закрываем файл

}

**Пример работы программы**



**Вывод:** используя технологию процедурного программирования разработал программу обработки текстовых файлов с числом строк не менее пяти, каждая из которых содержит не более 80 символов, в соответствии с индивидуальным заданием. Разработанная программа удаляет последнюю букву слова из введенного текста.