МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра информационно-аналитических систем безопасности

имени Л.С. Берштейна

**ОТЧЕТ**

**По лабораторной работе №4**

По курсу «Основы АиП»

**На тему**

«Работа с файлами»

Вариант №4

Выполнил:

студент гр. КТбо1-8

Васильченко Н.С.

«\_\_\_» 2017 г.

Проверил:

Профессор ИКТИБ

кафедры ИАСБ

Беляков С.Л

«\_\_\_» 2017 г.

**Таганрог – 2017**

Оглавление

[Цель работы 3](#_Toc499241095)

[Вариант задания 4](#_Toc499241096)

[Алгоритм работы 5](#_Toc499241097)

[Примеры входных и выходных данных 6](#_Toc499241098)

[Заключение 7](#_Toc499241099)

[Код программы 8](#_Toc499241100)

# Цель работы

Цель данной лабораторной работы ознакомить студента с понятием файла данных, научить его правильно использовать это понятие в программах.

# Вариант задания

Написать программу, которая считывает текст из файла и выберет слова, количество букв "М" в которых будет максимально, и выводит их в другой файл.

# Алгоритм работы

1. Подключаем библиотеки для ввода/вывода, выделения памяти в динамическом массиве, задержки консоли, работы со строками.
2. В главной функции объявляем массив.
3. Открываем файл для чтения.
4. Если в файле ничего нет, то пишем об ошибке и завершаем программу.
5. Считываем из файла длину нашего текста (кол-во символов).
6. Динамически выделяем память.
7. Считываем все слова из файла.
8. Вызываем функцию
9. В ней сначала рассматриваем слова.
10. Слова из всех строк записываем в одну строку.
11. Создаём ещё один массив, в котором будут храниться все слова с максимальным количеством букв «м».
12. Проверяем каждое слово.
13. Если в нём есть буква «м», то увеличиваем счётчик.
14. Если содержится количество букв «м» больше чем в переменной (которая хранит максимальное кол-во «м»), то перезаписываем эту переменную на большее значение.
15. Обнуляем массив, в котором будут храниться все слова с максимальным количеством букв «м».
16. Добавляем это слово в массив.
17. Иначе если содержится количество букв «м» равное переменной (которая хранит максимальное кол-во «м»), то объединяем имеющийся массив с этим словом.
18. Открываем файл для записи.

Записываем в него слова с максимальным кол-ом букв «м».

# Примеры входных и выходных данных

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| juy  mmjui  mm mm | mmjui mm mm |

# Заключение

В данной работе мы познакомились со средой создания сложных символьных массивов, научились обрабатывать строки. Были изучены различные методы ввода и вывода информации, вызова функций. Научились открывать для чтения и записи файлы, создавать файлы с текстом, и выводить его содержимое в другой по определённым требованиям.

# Код программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <conio.h>

#include <cstring>

void SearchWords(char \*words) // получение указателя на массив

{

char \*p; // указатель на массив

for (int i = 0; i<strlen(words); i++) // в цикле рассматривает все слова

{

if (words[i] == '\n') // проверят по строкам

{

words[i]=' '; // если встречаем символ перехода на новую строку (\n) заменяем на пробел.

} // То есть все слова из разных строк записываются в одну

}

p = strtok(words, " "); // разделение на слова, слово из массива записали в (p)

int MaxM = 0; // каксимальное к-во "М"

char WordM[100] = {}; // слова с каксимальным к-вом "М"

while (p != NULL) // пока не законяатся слова

{

int count = 0; // колличество M в слове

for (int i = 0; i<strlen(p); i++) //поиск в слове "М"

{ //

if ((p[i] == 'm') || (p[i] == 'M')) //

{ //

count += 1; // если найдено м, то увеличить счётчик

} //

} //

if (count != 0)

{

if (count > MaxM) //

{ //

MaxM = count; //если количество м в слове больше максимального, то сделать его максимальным и записать это слово

\*WordM = {}; // обнулили массив

strcat(WordM, p); //

strcat(WordM, " "); //

}

else if (count == MaxM) // если количество м совпадает с максимальным то добавить слово к существующему

{

strcat(WordM, p);

strcat(WordM, " ");

}

}

p = strtok(NULL, " "); // перейти к следующему слову

}

FILE \*Output = fopen("C:/Users/Никита/source/repos/LAB 4 - 4/Debug/output.txt", "w"); // создали указатель и открыли файл на запись

fprintf(Output, "%s", WordM); // записали в файл слова с максимальным кол-ом М

fclose(Output); // закрыли указатель

}

int main()

{

char \*words; // указатель на массив

int TextLength = 0; // длина всего текста(кол-во символов)

FILE \*Inp = fopen("C:/Users/Никита/source/repos/LAB 4 - 4/Debug/input.txt", "r"); // создали указатель и открыли файл для чтения

if (Inp == NULL) // если в файле ничего нет, то выдаст сообщение "нет данных"

{

printf("No data!");

\_getch();

return 0;

}

fscanf(Inp, "%d", &TextLength); // считывает количество символов из файла

words = (char\*)malloc((TextLength) \* sizeof(char)); // выделение памяти на массив

fread(words, sizeof(char), TextLength, Inp); // считывает все слова из файла

fclose(Inp); //закрывает указатель

SearchWords(words); // вызов функции

return 0;

}