



Отчет по лабораторной работе № 9 по курсу Фундаментальная информатика

Студент группы М8О-104Б-22 Ляпин Иван Алексеевич, № по списку 00

Контакты www, e-mail, icq, skype shad0w2020@mail.ru

Работа выполнена: « 7 » ноября 2022 г.

Преподаватель: асп. каф. 806 Потенко М.А.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202 __ г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа.

2. **Цель работы:** Составить программу на языке Си, выполняющую анализ и обработку вводимого текста.

3. **Задание (вариант № 3):** Подсчитать количество слов в комментариях в программе на Паскале.

4. **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Apple M1 Pro с ОП 16384 Мб, НМД 524 288 Мб. Монитор Liquid Retina XDR

Другие устройства _____

5. **Программное обеспечение (лабораторное):**

Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства macOS, наименование macOS Monterey версия 12.3

интерпретатор команд zsh версия 2.12.5

Система программирования C версия _____

Редактор текстов nano версия _____

Утилиты операционной системы Терминал _____

Прикладные системы и программы Xcode

Местонахождение и имена файлов программ и данных /Users/ivan/Desktop _____

6. Идея, метод, алгоритм решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Создадим файл "input.txt", куда будем записывать код программы на Паскале.

Затем будем посимвольно считывать элемент из файла до тех пор, пока текст в файле не закончится.

Так как нам необходимо исследовать элементы внутри двух конструкций комментария - очевидно, что все необходимые операции будем производить непосредственно встретив данные конструкции.

Будем фиксировать начало комментария и загонять программу в цикл, пока комментарий не закончится.

Если встретили букву в строке - фиксируем счетчик начала слова, также прибавляем счетчик слов на единицу.

Следует учесть проверку конструкции на нахождение внутри строки, заключённой в кавычки.

Также не забудем факт существования незакрытого комментария(то есть отсутствие закрывающего элемента конструкции).

В конце концов выводим количество подсчитанных слов.

7. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике [можно на отдельном листе] и тесты либо соображения по тестированию)

1)Создадим файл "input.txt" для записи кода на Паскале

2)Откроем файл с разрешением на чтение, а также выполним проверку на возможность открытия файла.

3)Введем следующие переменные для дальнейших операций:

"count_words" - счетчик найденных слов

"word_search" - булева переменная, отвечающая за факт встречи слова внутри комментария

"comment" - булева переменная, отвечающая за факт нахождения внутри комментария.

"с" - считываемый символ из файла.

4)Создадим цикл, выполняющийся пока файл не "станет" пустым.

5)Создадим проверку на присутствие конструкции комментария внутри кавычек

5.1)Если находится, то просто пропускаем все, что внутри, пока кавычки не закроются.

5.2)Если же комментарий не присутствует внутри, то пишем два условия на поиск начала конструкции комментария.

6)Если конструкция найдена, то присваиваем значение "1" переменной "comment" и входим в цикл, пока это значение не обратится в ноль.

7)Считываем элементы в цикле.

8)Если встретили букву(с помощью функции "isalpha"), при этом до неё нет букв, то увеличиваем счетчик кол-ва слов на единицу("count_words") и присваиваем значение "1" переменной "word_search". 9)Если считываемый элемент не является буквой(то есть "!isalpha"), то присваиваем значение "0" переменной "word_search" (слово закончено или даже еще не найдено, или является любым другим элементом)

10)Если комментарий не закрыли, то в конце обработки файла comment = 0 ; word_search = 0

11)Если встретили конец комментария, то присваиваем значение "0" переменной "comment".

12)В конце программы выводим значение счетчика

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем)

КОД

ТЕСТЫ

```
/*
    Лапин Иван Алексеевич
    И80-1046-22
    Вариант №27
*/

#include <math.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <string.h>
#include <stdbool.h>

#define FALSE 0
#define TRUE 1

int main(int argc, const char * argv[])
{
    FILE *file; // "создаем" файл
    file = fopen("input.txt", "r"); // открываем его с разрешением на чтение

    if (file == NULL) { // проверка на возможность открытия файла
        printf("Не удалось открыть файл!");
    }

    int count_words = 0; // вводим счетчик
    bool word_search = 0; // вводим переменную, отвечающую за нахождение слова в комментарии
    bool is_comment = FALSE; // вводим переменную, отвечающую за открытия комментария
    char c; // вводим переменную, что будет являться символом из файл

    while (!feof(file)) { // создаем цикл, выполняющийся пока файл не будет пустым
        while ((c = fgetc(file)) != EOF) {
            if (c != '\n') { // проверяем, находится ли конструкция внутри строки
                if (c == '{') { // встретили начало комментария
                    is_comment = TRUE; // bool переменная равна 1, так как открылся комментарий

                    while (is_comment) { // цикл пока не закроется комментарий

                        c = fgetc(file);
                        if (isalpha(c) && word_search == 0) { // если встретили букву, то поднимаем флаг, чтобы считать только первую букву слова
                            count_words += 1; // увеличиваем счетчик слов
                            word_search = 1; // поднимаем флаг
                        } else if (c == '}') { // условие закрытия комментария, опускаем флаг
                            is_comment = 0;
                            word_search = 0;
                        } else if (!isalpha(c)) { // если встречаем все кроме букв, то не учитываем
                            word_search = 0; // опускаем флаг(конец слова)
                        }
                        if (c == EOF) { // если забыли закрыть комментарий
                            is_comment = 0;
                            word_search = 0;
                        }
                    }
                } else if (c == '(') { // встретили первый элемент конструкции "("
                    c = fgetc(file); // считываем следующий элемент
                    if (c == '*') { // если он равен "*", то конструкция найдена

                        is_comment = TRUE; // открыт комментарий!

                        while (is_comment) { // цикл пока он не закроется
                            c = fgetc(file);

                            if (isalpha(c) && word_search == 0) {
                                count_words += 1;
                                word_search = 1;
                            } else if (c == '*') { // если встретили "*", проверяем на часть конструкции закрытия комментария

                                c = fgetc(file);

                                if (c == ')') { // конец комментария
                                    is_comment = FALSE;
                                    word_search = 0;
                                }
                            } else if (!isalpha(c)) {
                                word_search = 0;
                            }
                        }
                        if (c == EOF) { // если забыли закрыть комментарий
                            is_comment = FALSE;
                            word_search = 0;
                        }
                    }
                }
            }
        }
    }

    } else { // если конструкция внутри строки, то она уже не является комментарием, потому считать её мы не будем
        c = fgetc(file);
        while (c != '\n') {
            c = fgetc(file);
        }
    }

    }

    printf("%d\n", count_words); // выводим значение счетчика
    fclose(file); // закрываем файл, в котором работали
    return 0;
}
```

```
ivan@MacBook-Pro-Ivan labs % cat input.txt
var
    numDel, i, j: longint;
    d: array[1..2] of longint; (* we wewe w * fd fd *)
begin
    for i := 174457 to 174505 do begin
        numDel := 0;
        for j := 2 to i div 2 do begin (* one two three sec ) ds lol *)
            if i mod j = 0 then begin
                numDel := numDel + 1;
                if numDel > 2 then break;
                d[numDel] := j;
            end;
            if numDel = 2 then writeln(d[1], ' ', d[2]);
        end;
    end.
ivan@MacBook-Pro-Ivan labs % gcc main.c
ivan@MacBook-Pro-Ivan labs % ./a.out
6
ivan@MacBook-Pro-Ivan labs % cat input.txt
var
    numDel, i, j: longint;
    d: array[1..2] of longint; (* we wewe w * fd fd *)
begin
    for i := 174457 to 174505 do begin
        numDel := 0;
        for j := 2 to i div 2 do begin (* one two three sec ) ds lol *)
            if i mod j = 0 then begin
                numDel := numDel + 1;
                if numDel > 2 then break;
                d[numDel] := j;
            end;
            if numDel = 2 then writeln(d[1], ' ', d[2]);
        end;
    end.
ivan@MacBook-Pro-Ivan labs % ./a.out
11
ivan@MacBook-Pro-Ivan labs % cat input.txt
var
    numDel, i, j: longint;
    d: array[1..2] of longint; {}
begin
    for i := 174457 to 174505 do begin
        numDel := 0;
        for j := 2 to i div 2 do begin (**)
            if i mod j = 0 then begin
                numDel := numDel + 1;
                if numDel > 2 then break;
                d[numDel] := j;
            end;
            if numDel = 2 then writeln(d[1], ' ', d[2]);
        end;
    end.
ivan@MacBook-Pro-Ivan labs % ./a.out
0
ivan@MacBook-Pro-Ivan labs % cat input.txt
var
    numDel, i, j: longint;
    d: array[1..2] of longint; {*****}
begin
    for i := 174457 to 174505 do begin
        numDel := 0;
        for j := 2 to i div 2 do begin (*3333333*)
            if i mod j = 0 then begin
                numDel := numDel + 1;
                if numDel > 2 then break;
                d[numDel] := j;
            end;
            if numDel = 2 then writeln(d[1], ' ', d[2]);
        end;
    end.
ivan@MacBook-Pro-Ivan labs % ./a.out
0
ivan@MacBook-Pro-Ivan labs % █
```

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы: _____

11. Выводы: В данной лабораторной работе я смог составить программу, способную обрабатывать содержимое текста, тем самым совершать все необходимые операции. Стоит отметить немаловажную роль функций "isalpha"(элемент - буква) , "isdigit"(элемент - число) , "isspace"(элемент - пробел) и другие. Благодаря использованию данных функций, можно оптимизировать код, да и облегчить себе выполнение работы. Также я научился работе с файлами, а точнее их чтение в языке Си.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента

A stylized handwritten signature in black ink, consisting of several sharp, angular strokes.