**Отчет по лабораторной №6** по курсу«Фундаментальная информатика»

Студент группы М8О-103Б-23, № по списку 16

*Махмутов Дэниз*

Контакты e-mail: lustmodesta[@gmail.com](mailto:nikitaslobodin05@gmail.com)

Работа выполнена 26 декабря 2023г.

Преподаватель *Никулин С. П*., каф. 806

Отчет сдан \_\_\_\_\_\_\_\_г., итоговая оценка \_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Тема:** Обработка последовательности литер входного текста. Простейшие приемы лексического анализа.

**2. Цель работы:** Составление программы на Си для анализа входного текста

**3. Задание:**



**4. Оборудование** *ПЭВМ студента*: ПК, процессор AMD Ryzen 5 5600x, объем ОП: 16GB

**5. Программное обеспечение:**

Windows 10 WSL: Ubuntu

Distributor ID: *Ubuntu*

Description: *Ubuntu 22.04.3 LTS*

Release: *22.04*

Редактор текстов: *txt*

Утилиты ОС: *Терминал*

**6. Идея, метод, алгоритм решения задачи:** Посимвольно проходимся по всему входному тексту. Если символ является цифрой, то умножаем предыдущее значение number на 10 и добавляет к нему значение текущей цифры. - Если символ является пробелом, код выводит на экран последнее распознанное число, если оно больше INT\_MAX , и сбрасывает значение числа (`number`) обратно в -1.

**7. Сценарий выполнения работы**

1. Подключает необходимые заголовочные файлы: - stdio.h: для работы с функциями ввода/вывода - ctype.h: для работы с функциями проверки и преобразования символов - limits.h: для использования максимального значения типа int

2. Задает переменные: - c: для хранения введенного символа - number: для хранения числа, составленного из последовательности цифр

3. Начинает бесконечный цикл while (1), который будет выполняться до тех пор, пока не будет введен конец файла (EOF).

4. Считывает символ с помощью функции getchar() и присваивает его переменной c.

5. Проверяет, является ли считанный символ концом файла. Если да, то выходит из цикла с помощью ключевого слова break.

6. Проверяет, является ли считанный символ цифрой. Если да, то выполняется следующее: - Проверяет, было ли предыдущее число -1. Если да, то присваивает переменной number значение 0. - Умножает переменную number на 10 и добавляет к ней значение считанной цифры, полученное путем вычитания кода символа '0'.

7. Проверяет, является ли считанный символ пробелом или символом новой строки. Если да, то выполняется следующее: - Проверяет, является ли число number больше или равным максимальному значению типа int (INT\_MAX), и при этом не равно -1. Если да, то выводит число number с помощью функции printf(). - Присваивает переменной number значение -1, чтобы начать новое число.

8. Если считанный символ не является цифрой, пробелом или символом новой строки, то выполняется следующее: - Проверяет, было ли предыдущее число отличным от -1. Если да, выводит число number с помощью функции printf(). - Присваивает переменной number значение -1, чтобы начать новое число.

Допущен к выполнению работы. **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**8. Распечатка протокола**

#include <stdio.h>

#include <ctype.h>

#include <limits.h>

#define ENTER 10

#define POINT 46

#define TAB 9

int main() {

    int fl;

    int out = 0;

    fl = 0;

    int c;

    long long number = -1;

    while (1) {

        c = getchar();

        if (c == EOF){

            break;

        }

        if (isdigit(c)) {

            if (number == -1) {

                if (fl == 1){

                    putchar(c);

                    out = 1;

                    continue;

                }

                else{

                number = 0;

                }

            }

            if (isalpha(c)){

                fl = 1;

            }

            number = number \* 10 + (c - '0');

        } else if (isspace(c) || c == TAB || c == POINT || c == ENTER) {

            if ((number > INT\_MAX) && (number != -1) && (isspace(c) || c == TAB || c == POINT)) {

                printf("%lld", number);

                out = 1;

            }

            if (c == ENTER){

                if (out == 1){

                printf(" - Вывод\n");}

            }

            else{

                putchar(c);

                out = 1;

            }

            fl = 0;

            number = -1;

        }

        else {

            if ((number != -1)) {

                printf("%lld", number);

            }

            fl = 1;

            putchar(c);

            out = 1;

            number = -1;

        }

    }

    return 0;

}

(Где не подписано «Вывод» - ввод)

unix@DESKTOP-MPQDBS2:~/labs$ gcc c3.c

unix@DESKTOP-MPQDBS2:~/labs$ ./a.out

9

a9

a9 - Вывод

A9(((999999

A9(((999999 - Вывод

21474836471

21474836471 - Вывод

2147483647A

2147483647A - Вывод

21474836

21474836471 21474836475

21474836471 21474836475 - Вывод

2147483647 21474836475

21474836475 - Вывод

21474836471A 21474836475B

21474836471A 21474836475B - Вывод

21474b99999a

21474b99999a - Вывод

214748b999a 99

214748b999a - Вывод

**9. Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы. Программа выполнилась без ошибок.

**10 Замечание автора по существу работы:**

**11. Выводы** Проделав работу, я написал программу на Си для анализа входного текста.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_