**Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”**

Студент группы M8O-106Б-20 Почечура Артемий

Контакт e-mail: carbo59@yandex.ru

Преподаватель каф.806: Дубинин Алексей Владимирович

Входной контроль знаний с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан\_\_\_\_\_\_\_ <<\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_>> 2020г., итоговая оценка\_\_\_\_\_

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

**1. Тема:** Алгоритмы и структуры данных.

**2. Цель работы:** Составить программу на си, которая выполняет анализ и обработку вводимого

текста в соответствии с выданным вариантом.

**3. Задание (вариант №):** Вариант 6: подсчитать количество слов -  правильных идентификаторов языка Си.

**4. Оборудование ПЭВМ студента:** ЭВМ MSI GL62M 7RDX, процессор Intel core I7 c ОП 8192 Мб

**5. Программное обеспечение ЭВМ студента:**

Операционная система семейства Unix, наименование Ubuntu, версия 20.04

Интерпретатор команд GNU Bash версия 4.4.20

Редактор текстов nano

**Утилиты операционной системы: -**

**6. Идея, метод, алгоритм:**

* Реализовать функции, которые проверяют, является ли знак: разделителем (функция is\_sep), латинской буквой (функция is\_letter), цифрой (функция is\_digit), нижним подчёркиванием (функция is\_underline). Проверить работу каждой из них;
* Создать тип переменных с помощью typedef enum , в котором хранятся ассоциативные значения для переменных, которые используются при написании программы (таким образом “читать” код будет легче);
* Собрать функцию main, используя побочные функции и созданный тип, в которой будет реализовано выполнение поставленной задачи.

**7. Сценарий выполнения работы:**

Проверка работы функции is\_sep:

Входные данные: ‘ ‘;

Выходные данные: 1;

Входные данные: a;

Выходные данные: 0;

Проверка работы функции is\_letter:

Входные данные: 6;

Выходные данные: 0;

Входные данные: a;

Выходные данные: 1;

Входные данные: B;

Выходные данные: 1;

Проверка работы функции is\_digit:

Входные данные: a;

Выходные данные: 0;

Входные данные: 6;

Выходные данные: 1;

Проверка работы функции is\_underline:

Входные данные: \_;

Выходные данные: 1;

Входные данные: a;

Выходные данные: 0;

Проверка работы алгоритма:

root@DESKTOP-5HM2HTK:~# gcc -std=c99 -pedantic -Wall prog.c

root@DESKTOP-5HM2HTK:~# ./a.out

9asd asd\_sad a.a sda d a0

4

**8. Распечатка протокола:-**

**9. Дневник отладки: -**

**10. Замечание автора по существу работы: -**

**11. Выводы:** С помощью typedef enum можно создать собственный тип переменных, при его использовании легче ориентироваться в коде. При осуществлении ввода до EOF нужно делать проверку на соответствие поставленным условиям последнего слова, чтобы учесть все введённые слова (т.к. после слова может не оказаться разделителя).

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