

Отчет по лабораторной работе № 13 по курсу_Информатика_____

Студент группы: М8О-101Б-22, Шляхтуров Александр Викторович, № по списку: 26, Контакты

shliakhturov@gmail.com Работа выполнена: «2» декабря 2022г.

Преподаватель: каф. 806 Крылов Сергей Сергеевич, Входной

контроль знаний с оценкой _____

Отчет сдан « 15 » _____ 12 _____ 2022 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Программирование на языке СИ.
2. **Цель работы:** Составить программу на языке си, выполняющую обработку строки из файла.
3. **Задание** Вар 26. Определить, есть ли слово, все согласные которого свистящие (з, с, ц)

4. Оборудование

Оборудование ПЭВМ студента, если использовалось:

Процессор AMD Ryzen 5 5600, 6 ядер с ОП 16 гб, ТТН 512 гб

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства Linux, наименование Ubuntu версия 20.04.5 интерпретатор команд bash версия 5.0.17(1).

Система программирования VSCodeSS

Редактор текстов nano версия 6.2

Утилиты операционной системы WinRar, Microsoft Word.

Прикладные системы и программы Ubuntu wsl, Clion, Google Chrome

Местонахождение и имена файлов программ и данных на домашнем компьютере /home/alexander

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

Программа работает следующим образом. Функция `to_index` возвращает число от 0 до 21 в зависимости от того, какую букву обрабатывает программа. Функция `rus_let` возвращает единицу, если обрабатываемый символ принадлежит русскому алфавиту, и 0, если буква в нем не содержится. Далее реализовано множество всех согласных букв русского алфавита и множество свистящих согласных букв. Далее с помощью функции `getchar` происходит посимвольное считывание текстового файла и запись в переменную `c1` кода этого символа. Если считываемый символ одновременно не является пробелом и русской буквой, то мы его пропускаем и идем дальше. Если до этого мы не в слове и теперь текущий символ - буква, то мы заходим в слово. Переменной `in_word` присваивается значение единица. Если мы находимся в слове, проверяем, является ли буква согласной. Если она согласная, то добавляем в множество "w" согласных букв текущего слова.

Если мы были в слове и текущий символ - пробельный, то мы выходим из слова и проверяем, если множество согласных текущего слова равно множеству свистящих согласных и есть хотя бы одна такая согласная, то выводим успешный ответ и завершаем программу. В противном случае обнуляем множество "w" согласных пройденного слова, и программа продолжается. Если Нашлось подходящее под условие слово, то программа завершается с выводом «В файле есть слово, все согласные которого свистящие», в противном случае выводится строка «В файле отсутствуют слова, в которых все согласные свистящие».

6. Сценарий выполнения работы

Закомпилируем терминальной командой `gcc lr13.c`

Откроем файл `c.out` с помощью команды `./c.out <data.txt`, используя перенаправление потоков ввода\вывода

Для редактирования кода используем редактор GEDIT

Открытие кода в редакторе осуществляется с помощью терминальной команды `gedit lr13.c`

7.

Входные данные	Выходные данные	Описание тестируемого случая
Корова	В файле отсутствуют слова, в которых все согласные свистящие	Одно слово из несвистящих согласных
Заяц	В файле есть слово, все согласные которого свистящие	Одно слово только со свистящими согласными
Кошка собака корова заяц	В файле есть слово, все согласные которого свистящие	Набор слов, одно из которых только со свистящими согласными, стоящее в конце
Заяц корова кошка	В файле есть слово, все согласные которого свистящие	Набор слов, одно из которых только со свистящими согласными, стоящее впереди
зцс	В файле есть слово, все согласные которого свистящие	Слово из только свистящих согласных
Корова кошка собака	В файле отсутствуют слова, в которых все согласные свистящие	Слова с согласными несвистящими
з	В файле есть слово, все согласные которого свистящие	Единичный символ – не свистящая согласная
а	В файле отсутствуют слова, в которых все согласные свистящие	Единичный символ – не свистящая согласная
Цапля	В файле отсутствуют слова, в которых все согласные свистящие	Слово, в котором есть как свистящие, так и несвистящие согласные
	В файле отсутствуют слова, в которых все согласные свистящие	Пустой файл с данными

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

```
#include <stdio.h>
#include <ctype.h>
/*
Номера букв русского алфавита в целом типе (сначала алфавит строчных, потом заглавных)
-80а
-79б
-78в
-77г
```

-76д
-75е
-111ё
-74ж
-73з
-72и
-71й
-70к
-69л
-68м
-67н
-66о
-65п
-128р
-127с
-126т
-125у
-124ф
-123х
-122ц
-121ч
-120ш
-119щ
-118ъ
-117ы
-116ь
-115э
-114ю
-113я
-112А
-111Б
-110В
-109Г
-108Д
-107Е
-127Ё
-106Ж
-105З
-104И
-103Й
-102К
-101Л
-100М
-99Н
-98О
-97П
-96Р
-95С
-94Т
-93У
-92Ф
-91Х
-90Ц
-89Ч
-88Ш

```

-87Щ
-86Ъ
-85Ы
-84Ь
-83Э
-82Ю
-81Я
*/
int to_index(int a) {
    if (a == -79 || a == -111) return 0;
    if (a == -78 || a == -110) return 1;
    if (a == -77 || a == -109) return 2;
    if (a == -76 || a == -108) return 3;
    if (a == -74 || a == -106) return 4;
    if (a == -73 || a == -105) return 5; // с
    if (a == -71 || a == -103) return 6;
    if (a == -70 || a == -102) return 7;
    if (a == -69 || a == -101) return 8;
    if (a == -68 || a == -100) return 9;
    if (a == -67 || a == -99) return 10;
    if (a == -65 || a == -97) return 11;
    if (a == -128 || a == -96) return 12;
    if (a == -127 || a == -95) return 13; // з
    if (a == -126 || a == -94) return 14;
    if (a == -124 || a == -92) return 15;
    if (a == -123 || a == -91) return 16;
    if (a == -122 || a == -90) return 17; // се
    if (a == -121 || a == -89) return 18;
    if (a == -120 || a == -88) return 19;
    if (a == -119 || a == -87) return 20;
    return 21;
}

// Проверка, на то, русская ли это буква
int rus_let(int a) {
    if (a <= -65 && a >= -128) return 1;
    return 0;
}

//Множество, состоящее только из гласных букв русского алфавита. Реализованно, как число, в
котором 1 и 0 на i-ом индексе (справа, нумерация с 1) в двоичном представлении числа
означают, что буква с i-ым номером по алфавиту присутствует или же отсутствует
соответственно.
#define SVIST (1u<<(5) | 1u<<(13) | 1u<<(17))
#define CONSONANTS (1u<<(0) | 1u<<(1) | 1u<<(2) | 1u<<(3) | 1u<<(4) | 1u<<(5) | 1u<<(6) |
1u<<(7) | 1u<<(8) | 1u<<(9) | 1u<<(10) | 1u<<(11) | 1u<<(12) | 1u<<(13) | 1u<<(14) |
1u<<(15) | 1u<<(16) | 1u<<(17) | 1u<<(18) | 1u<<(19) | 1u<<(20))
int main() {

    // с - текущий символ, который считываем; w - множество, из гласных букв текущего слова;
num - номер по алфавиту текущего слова; in_word - ключ, проверяющий, находимся ли мы в
слове; c1 - целочисленное представление буквы;
    char c;
    int c1;
    unsigned int w = 0, num, in_word = 0;

```

```

// Открытие файлов
FILE *file = fopen("data.txt", "r");

// Считывание файла
while ( (c = getc(file)) != EOF ){
    c1 = (int) c;

    // Если не буква и не пробел, пропускаем
    if (!(rus_let(c1) || isspace(c))) {
        continue;
    }

    // Если сейчас мы не в слове и текущий символ - буква, то мы заходим в слово.
    if (in_word == 0 && rus_let(c1)) {
        //printf("%d\n", c1);
        in_word = 1;
        w = 0;
    }

    // Если мы были в слове и текущий символ - пробельный, то мы выходим из слова и
    проверяем, если множество согласных текущего слова равно множеству свистящих согласных и
    есть хотя бы одна такая согласная, то выводим успешный ответ и завершаем
    программу.

    if (in_word && isspace(c)) {
        if ( (w | SVIST) == SVIST && w) {
            printf("В файле есть слово, все согласные которого свистящие\n");
            return 0;
        }
        in_word = 0;
        w = 0;
        continue;
    }

    // Если мы находимся в слове, проверяем, является ли буква согласной. Если она
    согласная, то добавляем в множество согласных букв текущего слова
    if (in_word) {
        num = 1u<<(to_index(c1));

        if ( (num | CONSONANTS) == CONSONANTS) {
            w |= num;
        }
    }

}

// Если ответ об успешном завершении программы не был выведен, значит не было подходящих
слов

```

```
printf("В файле отсутствуют слова, в которых все согласные свистящие\n");

// Заккрытие файлов
fclose(file);
}
}
```

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 других 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**

11. **Выводы** В ходе работы была составлена программа, обрабатывающая строку символов из входного потока. Была реализована работа с множествами на языке СИ, что является для меня новым навыком.

Подпись студента _____