

# Отчет по лабораторной работе № 13 по курсу “Фундаментальная информатика”

Студент группы М80-101Б-22, Бычков Артур Сергеевич, № по списку 2

Контакты email: bychkovarthur@gmail.com

Работа выполнена: «26» ноября 2022г.

Преподаватель: каф. 806 Крылов Сергей Сергеевич

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

Отчет сдан « » \_\_\_\_\_ 202 \_\_\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Множества
2. **Цель работы:** В соответствии с вариантом задания составить программу проверки характеристик введенных последовательностей слов и печати развернутого ответа.

3. **Задание** (вариант № 2): Определить, есть ли слово, все согласные которого различны.

4. **Оборудование:**

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD Ryzen 5 5500U 2.10 GHz, 6 ядер с ОП 8192 Мб, ТТН 512000 Мб. Мониторы Lenovo.

5. **Программное обеспечение:**

*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 Clion версия 2021.1.3

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/artur

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)
  1. Составим "алфавит" только из согласных букв ("алфавит" реализуется, как число, в котором 1 и 0 на  $i$ -ом индексе (справа налево, нумерация с 1) в двоичном представлении числа означают, что буква с  $i$ -ым номером по алфавиту присутствует или же отсутствует соответственно)
  2. Создать `good`, которое при 1 означает, что слово подходит (в нем нет повторяющихся согласных) и если 0, то не подходит. И создать состояние `in_word`, при 1 означает, что мы находимся в слове, иначе, если `in_word = 0`, то мы находимся вне слова. Создать число `letters`, которое работает, как алфавит из 1) пункта, только в нем содержатся буквы, из текущего слова.
  3. Понимать только считывать файл, переводя все буквы в нижний регистр.
  4. Если встречается буква и состояние `in_word = 0`, то мы вошли в слово. Переключаем состояния `in_word` и `good` на единицу, а алфавит слова сбрасываем до нуля.
  5. Проверяем каждую букву на то, является ли она согласной:
    1. В переменную `num` записываем число, равное 2 в степени порядковый буквы по алфавиту – 1, т.е. буква  $a = 1$ ,  $b = 2$ ,  $z = 2^{25} = 33\ 554\ 432$ .
    2. Проверяем, является ли буква согласной: от числа, которое является «алфавитом» только из согласных букв, берем остаток, на  $(2 * num)$ . В итоге, если рассматривать двоичную запись числа, после этих действий останется число, где старший разряд – означает, есть ли текущая буква в «алфавите». (Это можно проверить, сравнив с получившийся остаток с `num`. Если он больше либо равен `num`, то это означает, что эта буква есть в «алфавите»)
    3. Если буква является согласной, такое же действие проделываем уже не со всем «алфавитом» согласных, а уже с «алфавитом» текущего слова. Если эта согласная уже была в слове, значит такое слово нам не подходит и состояние `good` переключаем на 0.
    4. В «алфавит» текущего слова, прибавляем текущую букву.
  6. Если встречаем пробельный символ (табуляция, пробел, перенос строки), и состояние `in_word = 0`, то выходим из слова. Переключаем `in_word` на 0, а также, если состояние `good` было равно 1, то увеличиваем счетчик подходящих слов, а `good` сбрасываем до нуля.
  7. Выводим ответ и закрываем файлы.

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

<i><b>Ввод</b></i>	<i><b>Вывод</b></i>
	Нет подходящих слов
Hello	Нет подходящих слов
Hello world	Подходящих слов - 1
abvgd afdas	Подходящих слов - 2
abvgd afdass	Подходящих слов - 1
abvgdd afdass	Нет подходящих слов
Generally, any device that can perform numerical calculations, even an adding machine, may be called a computer but nowadays this term is used especially for digital computers. Computers that once weighed thirty tons now may weigh as little as one point eight kilograms. Microchips and microprocessors have...	Подходящих слов - 35

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

```
artur@artur-VirtualBox:~$ cat lab13.c
#include <stdio.h>
#include <ctype.h>
```

/\* Лабораторная работа №13. Вариант 2

Определить, есть ли слово, все согласные которого различны.

Студент группы М8О-101Б-22 Бычков А.\*/

```
// "Алфавит" только из согласных букв ("алфавит" реализуется, 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 CONSONANTS (1u<<('q' - 'a') | 1u<<('w' - 'a') | 1u<<('r' - 'a') | 1u<<('t' - 'a') | 1u<<('p' - 'a') | 1u<<('s' - 'a') |
1u<<('d' - 'a') | 1u<<('f' - 'a') | 1u<<('g' - 'a') | 1u<<('h' - 'a') | 1u<<('j' - 'a') | 1u<<('k' - 'a') | 1u<<('l' - 'a') | 1u<<('z' - 'a') |
1u<<('x' - 'a') | 1u<<('c' - 'a') | 1u<<('v' - 'a') | 1u<<('b' - 'a') | 1u<<('n' - 'a') | 1u<<('m' - 'a'))
```

```
int main() {
    // c - текущий символ; letters - "алфавит" из букв текущего слова; num - номер по алфавиту текущей буквы;
in_word - ключ, проверяющий, находимся ли мы в слове; good - ключ, проверяющий, подходит ли слово; count -
счетчик подходящих слов.
```

```
    char c;
    unsigned int letters, num, in_word, good, count = 0;
```

```
    // Открытие файлов
```

```
    FILE *file = fopen("lab13_data.txt", "r");
```

```
    FILE *ans = fopen("lab13_answer.txt", "w");
```

```
    // Чтение файла
```

```
    while ( (c = getc(file)) != EOF ){
```

```
        // Не учитываем регистр
```

```
        c = tolower(c);
```

```
        if (c == '.' || c == ',') {
            continue;
        }
```

```
        // Заходим в слово, если это буква, и до этого было не слово
```

```
        if (isalpha(c) && in_word == 0) {
            in_word = 1;
            good = 1;
            letters = 0;
        }
```

```
        // Выходим из слова, если это пробельный символ (табуляция, пробел, перенос строки), и если до этого
был в слове
```

```
        if (isspace(c) && in_word) {
```

```
            // Если слово подходящее, увеличиваем счетчик
```

```
            if (in_word == 1 && good) {
                good = 0;
                count++;
            }
```

```
            in_word = 0;
```

```
            continue;
```

```
        }
```

```
        // Находимся внутри слова
```

```
        if (in_word) {
```

```
            // Номер буквы по алфавиту
            num = 1u<<(c - 'a');
```

```
            // Проверяем, является ли согласной
```

```
            if ( ( CONSONANTS % (2 * num) ) >= num ){
```

```
                // Проверяем, является ли согласной, которая уже была в этом слове, если она таковая,
то все слово не подходит
```

```
                if (( letters % (2 * num) ) >= num){
                    good = 0;
                }
```

```
            }
```

```

        // В "алфавит" записываем букву, которая попалась
        letters |= num;

    }
}
// Вывод ответа
if (count) {
    fprintf(ans, "Подходящих слов - %d\n", count);
} else {
    fprintf(ans, "Нет подходящих слов");
}

// Заккрытие файлов
fclose(file);
fclose(ans);
}
artur@artur-VirtualBox:~$ cat lab13_data.txt
More and more, the operations of our businesses, governments, and financial institutions are controlled by information
that exists only inside computer memories. Anyone clever enough to modify this information
artur@artur-VirtualBox:~$ ./a.out
artur@artur-VirtualBox:~$ cat lab13_answer.txt
Подходящих слов - 18

```

**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	Вре мя	Событие	Действие по исправлению	Примечание

**10. Замечания автора** по существу работы

#### 11. Выводы

Я научился составлять программы, написанные на языке С, для проверки характеристик введенных последовательностей слов и печати развернутого ответа, используя множества.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента

---