

Отчет по лабораторной работе № 7

по курсу Фундаментальная информатика

Студент группы М8О-111Б-23 Бугренков Владимир Петрович,
№ по списку 4, Контакты e-mail: vladimir.bugrenkov@yandex.ru

Работа выполнена: «11» декабря 2023 г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Отчет сдан «12» декабря 20 23 г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Множества

2. Цель работы: Научиться работать с множествами.

3. Задание (вариант № 4): Есть ли слово, хотя бы одна согласная которого повторяется

4. Оборудование (лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб, НМД _____ Мб.

Терминал _____ адрес _____, Принтер _____

Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор AMD Ryzen 5 с ОП 16 Гб НМД 512 Гб. Монитор 1920x1080~60Hz

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 Linux, наименование Ubuntu версия 22.04.2

интерпретатор команд GNU bash версия 5.1.16.

Система программирования C.

Редактор текстов emacs версия 29.1

Утилиты операционной системы _____

Прикладные системы и программы Emacs

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица])

Создадим множество согласных латинских букв. Создадим функцию перевода символа в множество, состоящее из только вводимой буквы. Создадим флаг со значением 0. Создадим пустое множество `word_consonants_set` – множество всех согласных букв слова

Цикл будет считывать символы, пока не встретит конец файла EOF.

Если вводимый символ является разделителем, то обнуляем множество `word_consonants_set`.

Если символ является согласной буквой, то проверяем `word_consonants_set` на равенство нулю, если равенство не достигается, то добавляем вводимую согласную букву в множество `word_consonants_set`, иначе проверяем равенство `word_consonants_set` и `word_consonants_set` объединенный с текущей согласной буквой, если равенство достигается, то значит в слове уже оказалась буква равная текущей вводимой и, следовательно, нам это подходит и флагу `flag` присваиваем значение 1, иначе идем к следующему символу

Если символ не является согласной буквой, то идем к следующему символу
После цикла: Если `flag` равен 1, то выводим, что нашлось нужное слово, иначе выводим, что не нашлось!

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

Тесты

Ввод	Вывод
AbC, EFGH, iJkLm, NoPqR, StU, VwX,Y z,	НЕ нашлось слова, в котором хотя бы одна согласная повторяется!
Hello, this is the sentence with repetitive CONSONANTS letters!	Нашлось слова, в котором хотя бы одна согласная повторяется!
abc, , aeeed, eeed, aaaa, lp	НЕ нашлось слова, в котором хотя бы одна согласная повторяется!

	НЕ нашлось слово, в котором хотя бы одна согласная повторяется!
.....BBBBV.....	Нашлось слово, в котором хотя бы одна согласная повторяется!

Код программы:

```
#include <stdio.h>
#include <ctype.h>

//множество согласных букв
#define CONSONANTS (1u << ('b' - 'a') | 1u << ('c' - 'a') | 1u << ('d' - 'a') | 1u <<
('f' - 'a') | 1u << ('g' - 'a') | 1u << ('h' - 'a') | 1u << ('j' - 'a') | 1u << ('k' -
'a') | 1u << ('l' - 'a') | 1u << ('m' - 'a') | 1u << ('n' - 'a') | 1u << ('p' - 'a') |
1u << ('q' - 'a') | 1u << ('r' - 'a') | 1u << ('s' - 'a') | 1u << ('t' - 'a') | 1u <<
('v' - 'a') | 1u << ('w' - 'a') | 1u << ('x' - 'a') | 1u << ('z' - 'a'))

unsigned int char_to_set(char c) {
    c = tolower(c); //для упрощения кода не будем учитывать регистр
    if (c < 'a' || c > 'z') {
        return 0; // если это не буква, то 0
    } else {
        return 1u << (c - 'a');
    }
}

int main() {
    int c, flag = 0;
    unsigned int word_consonants_set = 0;
    unsigned int temp_consonants;
    while ((c = getchar()) != EOF) {
        if (c == '\n' || c == '\t' || c == ',' || c == ' '
            || c == '+' || c == '-' || c == '=' || c == '/' || c == '.'
            || c == '*' || c == '(' || c == ')' || c == ';' || c == '`') {
            word_consonants_set = 0;
            continue;
        }
        if ((char_to_set(c) & CONSONANTS) != 0) { // работаем только с согласными
            if (word_consonants_set == 0) {
                word_consonants_set = word_consonants_set | char_to_set(c);
            } else {
                temp_consonants = word_consonants_set;
                word_consonants_set = word_consonants_set | char_to_set(c);
                if (temp_consonants == word_consonants_set) {
                    flag = 1;
                }
            }
        }
    }
    if (flag == 1) {
        printf("\nНашлось слово, в котором хотя бы одна согласная повторяется!\n");
    }
}
```

```

} else {
    printf("\nНЕ нашлось слово, в котором хотя бы одна согласная повторяется!\n");
}
return 0;
}

```

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

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7\$ cat main.c

```
#include <stdio.h>
```

```
#include <ctype.h>
```

```
//множество согласных букв
```

```
#define CONSONANTS (1u << ('b' - 'a') | 1u << ('c' - 'a') | 1u << ('d' - 'a') | 1u << ('f' - 'a') | 1u << ('g' - 'a') | 1u << ('h' - 'a') | 1u << ('j' - 'a') | 1u << ('k' - 'a') | 1u << ('l' - 'a') | 1u << ('m' - 'a') | 1u << ('n' - 'a') | 1u << ('p' - 'a') | 1u << ('q' - 'a') | 1u << ('r' - 'a') | 1u << ('s' - 'a') | 1u << ('t' - 'a') | 1u << ('v' - 'a') | 1u << ('w' - 'a') | 1u << ('x' - 'a') | 1u << ('z' - 'a'))
```

```
unsigned int char_to_set(char c) {
```

```
    c = tolower(c); //для упрощения кода не будем учитывать регистр
```

```
    if (c < 'a' || c > 'z') {
```

```
        return 0; // если это не буква, то 0
```

```
    } else {
```

```
        return 1u << (c - 'a');
```

```
    }
```

```
}
```

```
int main() {
```

```
    int c, flag = 0;
```

```
    unsigned int word_consonants_set = 0;
```

```
    unsigned int temp_consonants;
```

```
    while ((c = getchar()) != EOF) {
```

```
        if (c == '\n' || c == '\t' || c == ';' || c == ''
```

```
            || c == '+' || c == '-' || c == '=' || c == '/' || c == '.'
```

```
            || c == '*' || c == '(' || c == ')' || c == ':' || c == '^') {
```

```
            word_consonants_set = 0;
```

```
            continue;
```

```
        }
```

```
        if ((char_to_set(c) & CONSONANTS) != 0) { // работаем только с согласными
```

```
            if (word_consonants_set == 0) {
```

```
                word_consonants_set = word_consonants_set | char_to_set(c);
```

```
            } else {
```

```

temp_consonants = word_consonants_set;
word_consonants_set = word_consonants_set | char_to_set(c);
if (temp_consonants == word_consonants_set) {
    flag = 1;
}
}
}
}
if (flag == 1) {
    printf("\nНашлось слово, в котором хотя бы одна согласная повторяется!\n");
} else {
    printf("\nНЕ нашлось слово, в котором хотя бы одна согласная повторяется!\n");
}
return 0;
}
qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$ gcc
main.c
qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$
./a.out
AbC, EFGH, iJkLm, NoPqR, StU, VwX,Y z,

```

НЕ нашлось слово, в котором хотя бы одна согласная повторяется!

```

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$
./a.out
Hello, this is the sentence with repetitive CONSONANTS letters!

```

Нашлось слово, в котором хотя бы одна согласная повторяется!

```

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$
./a.out
abc, , aeeed, eeed, aaaa, lp

```

НЕ нашлось слово, в котором хотя бы одна согласная повторяется!

```

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$
./a.out
,,,,,,,,,,,,,,,,

```

НЕ нашлось слово, в котором хотя бы одна согласная повторяется!

```

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$
./a.out
,,,,,,,,,BBBBB,,,,

```

Нашлось слово, в котором хотя бы одна согласная повторяется!

```

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Documents/Информатика/Laba7$ quit

```

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 по существу работы

11 Выводы

В ходе выполнения лабораторной работы, я научился работать с множествами на языке си.

Подпись студента Бугренков В.П.