Отчет по лабораторной работе № 13 по курсу «Фундаментальная информатика»

Студент группы М8О-109Б-22 Концебалов Олег Сергеевич

Контакты: telegram @baronpipistron

Работа выполнена: 15.12.2022

Преподаватель: каф.806 Сысоев Максим Алексеевич

Отчет сдан «19» декабря 2022 г., итоговая оценка ____

Подпись преподавателя _____

1. Тема: Множества

2. Цель работы: Написать программу на языке C, которая будет выполнять действия, указанные в задании

3. Задание (вариант № 19): Есть ли слово, содержащее одну гласную, возможно несколько раз

4. Оборудование (студента):

Процессор AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz, ОП 16,0 Гб, SSD 512 Гб. Монитор 1920x1080 144 Hz

5. Программное обеспечение (студента):

Операционная система семейства Linux, наименование Ubuntu, версия 18.10

Интерпретатор команд: bash, версия 4.4.19

Система программирования – версия --, редактор текстов Emacs, версия 25.2.2

Утилиты операционной системы –

Прикладные системы и программы –

Местонахождение и имена файлов программ и данных на домашнем компьютере –

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

Идея заключается в том, то мы с помощью метода getchar() считываем символы из потока стандартного ввода, и с помощью конструкции switch-case и 3 множеств (гласные, согласные и цифры) проверяем поступивший на вход символ и выполняем с ним определенные действия. Если символ является гласной, то записываем его во множество. После, когда встречаем разделитель, проверяем множество гласных слова – если в нем содержится только одна гласная, то слово содержит одинаковые гласные => выводим, что такое слово содержится и завершаем программу; если во множестве больше одного элемента, то обнуляем множество и переходим к следующему слову. Если слов с одинаковыми гласными нет, и мы дошли до конца строки, то выводим, что такого слова нет

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

1. Продумываю алгоритм, пишу отдельные его части на языке С
2. Пишу программу на языке С
3. Тестирую финальную версию программы и отлавливаю баги, если есть

Входные данные	**Выходные данные**	**Описание тестируемого случая**
a *	Yes, there is such a word	Строка, содержащая одну гласную
bcdghghgph *	No, there is no such word	Только согласные
BCAaadgT *	Yes, there is such a word	Заглавные и строчные буквы
12145 15aaa15rar *	Yes, there is such a word	Цифры и буквы
gxgxXXxgGtth4xgxg ague , FFoflolu amMe zXc dead inside zfoorka 155aaAaaAAArkr4b4 *	Yes, there is such a word	Все комбинации и есть подходящее слово
gxgxXXxgGtth4xgxg ague , FFoflolu amMe zXc dead inside zfoorka 155aaeuaaAAArkr4b4 *	No, there is no such word	Все комбинации и нет подходящего слова

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

```
#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <assert.h>

#include <inttypes.h>


const uint64_t vowels = (

    1u << ('a' - 'a') | 1u << ('e' - 'a') | 1u << ('i' - 'a') |

    1u << ('o' - 'a') | 1u << ('u' - 'a') | 1u << ('y' - 'a')

);


const uint64_t 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')
);

const uint64_t numbers = (
        1u << ('0' - '0') | 1u << ('1' - '0') | 1u << ('2' - '0') | 1u << ('3' - '0') | 1u << ('4' - '0') |
        1u << ('5' - '0') | 1u << ('6' - '0') | 1u << ('7' - '0') | 1u << ('8' - '0') | 1u << ('9' - '0')
);

int is_vowels(char symbol) {
    uint64_t symbol_num = 1u << (symbol - 'a');
    if ((symbol_num & ~vowels) == 0){
        return 1;
    }
    return 0;
}

int is_consonants(char symbol) {
    uint64_t symbol_num = 1u << (symbol - 'a');
    if ((symbol_num & ~consonants) == 0){
        return 1;
    }
    return 0;
}

int is_numbers(char symbol) {
    uint64_t symbol_num = 1u << (symbol - '0');
    if ((symbol_num & ~numbers) == 0){
        return 1;
    }
    return 0;
}
```

```
int check_word_set(uint64_t word_set){
    uint64_t array[6] = {1u << ('a' - 'a'), 1u << ('e' - 'a'), 1u << ('i' - 'a'),
                         1u << ('o' - 'a'), 1u << ('u' - 'a'), 1u << ('y' - 'a')};

    char array_vowels[6] = {'a', 'e', 'i', 'o', 'u', 'y'};

    int sum = 0;
    for(int i = 0; i < 5; ++i){
        if (array[i] == (word_set >> (array_vowels[i] - 'a'))){
            sum += 1;
        }
    }
    return sum;
}

void test_is_vowels(){
    assert(is_vowels('a') == 1);
    assert(is_vowels('i') == 1);
    assert(is_vowels('z') == 0);
    assert(is_vowels('x') == 0);
}

void  test_is_consonants(){
    assert(is_consonants('b') == 1);
    assert(is_consonants('v') == 1);
    assert(is_consonants('o') == 0);
    assert(is_consonants('e') == 0);
}

void test_is_numbers(){
    assert(is_numbers('1') == 1);
    assert(is_numbers('3') == 1);
```

```
    assert(is_numbers('b') == 0);
    assert(is_numbers('h') == 0);
}

void all_tests(){
    test_is_vowels();
    test_is_consonants();
    test_is_numbers();
}

int main(){
    all_tests();

    int symbol;
    int state = 1, count = 0, flag_1 = 0, flag_2 = 0;
    uint64_t word_set = 0;

    while ((symbol = tolower(getchar())) != '*'){
        switch (state) {
            case 1:
                if ((is_vowels((char)symbol) & vowels) == 1 && symbol != '5'){
                    word_set |= (1 << ((char)symbol - 'a'));
                    count += 1;
                    state = 1;
                }
                else if (((is_consonants((char)symbol) & consonants) == 1) || ((is_numbers((char)symbol) & numbers) == 1)){
                    state = 1;
                }
                else if (symbol == ' ' || symbol == ',' || symbol == '\n' || symbol == '\t'){
                    if (check_word_set(word_set) == 1){
                        flag_1 += 1;
```

```
            }
            else if (check_word_set(word_set) > 1){
                flag_2 -= 1;
            }
            if (count == 1 || (flag_1 != 0 && flag_2 == 0 && count > 0)){
                printf("Yes, there is such a word");
                exit(0);
            }
            word_set = 0;
            count = 0;
            state = 2;
        }
        break;


    case 2:
        flag_1 = 0;
        flag_2 = 0;
        if ((is_vowels((char)symbol) & vowels) == 1){
            word_set |= (1 << ((char)symbol - 'a'));
            count += 1;
            state = 1;
        }
        else if (((is_consonants((char)symbol) & consonants) == 1) || ((is_numbers((char)symbol) & numbers) == 1)){
            state = 1;
        }
        else if (symbol == ' ' || symbol == ',' || symbol == '\n' || symbol == '\t'){
            word_set = 0;
            count = 0;
            flag_1 = 0;
            flag_2 = 0;
            state = 2;
```

```
            }
            break;

        default:
            break;
    }
}

if (flag_1 == 0 || flag_2 < 0){
    printf("No, there is no such word");
}

return 0;
}
```

9. Дневник отладки *(дата 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*	*Примечания*
1	дом	14.12.2022	18:34	Не переходит по разделителям	Оказалось, что вообще не доходит до этого момента, пришлось искать ошибку	
2	дом	14.12.2022	19:44	Всегда выводит, что слова нет	Опять не входит в один из case	
3	дом	14.12.2022	21:12	Во время тестов вылезают единичные баги	Фиксим все	
4	дом	15.12.2022	17:15	Узнал, что нельзя использовать массив для отлова гласных	Переписываем с использованием множеств	Тильт
5	дом	15.12.2022	18:04	Не выполняется	Ошибся в задачи	

				функция для проверки слова на разные гласные	массивов в функции, минут 20 фиксил	

*Примечание: как и в 11 лабораторной для завершения ввода необходимо напечатать символ *. Не работает стандартное сочетание для EOF Ctrl+D, не знаю с чем связано. Получается маленький костыль*

10. Замечания автора *(по существу работы)*

Замечания отсутствуют

11. Вывод

От этой лабораторной остались смешанные впечатления. С одной стороны, выполнять ее, как и 11, было довольно-таки интересно и в какой-то мере сложно. Приходилось держать в голове одновременно много вещей, так же понравилось фиксить баги (удивительно), которые вылезали по мере тестов. Но, с другой стороны, работать со множествами мне не понравилось от слова совсем. Не могу даже сказать почему так, просто не мое. Но в целом от лабораторной положительных эмоций осталось больше (кроме того момента, когда узнал, что через массивы нельзя, и пришлось переписывать на множества)

Работа на 7/10

Подпись студента _____