

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

Студент группы М80-109Б-22 Фомин Иван Дмитриевич, № 22

Контакты email: grenka388@gmail.com,

Telegram: @Haliaven

Работа выполнена: «26» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «28» ____11____2022 г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Алгоритмы и структуры данных
2. **Цель работы:** Сделать задание на set.
3. **Задание:** Вариант 2. Есть ли слово, все согласные которого различны?
4. **Оборудование** (студента):

Процессор AMD Ryzen 7 4800HS with Radeon Graphics с ОП 8 Гб, SSD 512 Гб. Монитор 1920x1080

5. **Программное обеспечение** (студента):

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04 *jammy*

интерпретатор команд: *bash* версия 5.1.16(1)-release

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

6. Идея, метод, алгоритм

У нас есть число, в которое записан весь латинский алфавит (по одному биту на букву) и где поставлены 1 на всех местах, где находятся согласные.

Далее для каждой буквы мы проверяем, что, если при вычитании (то есть при логическом умножении отрицания) из этого числа переменной CONSONANTS получится ноль, то это число согласная, иначе – гласная.

Примерно такой же алгоритм для проверки на то, встречалась эта согласная в слове или нет.

7. Сценарий выполнения работы

Входные данные	Выходные данные	Описание тестируемого случая
Siluehflu wleiuhfliuwfwjhe fjhek wkejhfkewjfh ekjfhekh	0	Несколько неподходящих слов
d	1	Один символ
Bebra babus bro	1	Обычный случай

8. Распечатка протокола

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

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')
);

uint64_t addCharToNum(char c) {
    if (c > 'z' || c < 'a')
        return 0;
    return 1u << (c - 'a');
}

int isConsonant(char c) {
    uint64_t cNum = 1u << (c - 'a');

    if ((cNum & ~CONSONANTS) == 0) return 1;
    return 0;
}

int isAlreadyInNum(char c, uint64_t num) {
    uint64_t currentChar = 1u << (c - 'a');

    if ((currentChar & ~num) == 0) return 1;
    return 0;
}

void test_addCharToNum() {
    assert((addCharToNum('a') & ~(1u << ('a' - 'a')) == 0);
    assert((addCharToNum('b') & ~(1u << ('b' - 'a')) == 0);
    assert((addCharToNum('g') & ~(1u << ('g' - 'a')) == 0);
}

void test_isConsonant() {
    assert(isConsonant('a') == 0);
    assert(isConsonant('o') == 0);
    assert(isConsonant('k') == 1);
    assert(isConsonant('c') == 1);
}

void test_isAlreadyInNum() {
    assert(isAlreadyInNum('a', (1u << ('a' - 'a')) == 1);
    assert(isAlreadyInNum('b', (1u << ('a' - 'a')) == 0);
}

void executeTests() {
    test_addCharToNum();
    test_isConsonant();
    test_isAlreadyInNum();
}
```

```

int main() {
    executeTests();
    // Есть ли слово, где все согласные различны?
    // a b c d e f g h i j k l m n o p q r s t u v w x y z
    // 0 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 ----- CONSONANTS

    char c;
    char isInWord = 0;
    char hasUniqueConsonants = 1;
    char hasConsonants = 0;
    uint64_t currentWord = 0;

    while ((c = tolower(getchar())) != '1') {
        if (isalpha(c) == 0 || isdigit(c) == 1) {
            if (isInWord == 1 && hasUniqueConsonants == 1 && hasConsonants == 1) {
                printf("1");
                return 1;
            }

            hasUniqueConsonants = 1;
            hasConsonants = 0;
            isInWord = 0;
            currentWord = 0;
            continue;
        } else if (hasUniqueConsonants == 0) continue;
        isInWord = 1;

        if (isConsonant(c) == 1) {
            if (isAlreadyInNum(c, currentWord) == 1) {
                hasUniqueConsonants = 0;
                continue;
            }

            hasConsonants = 1;
            currentWord |= addCharToNum(c);
        }
    }

    printf("0");
    return 0;
}

```

9. Дневник отладки

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	26.11.2022	04:20	Вспомнил, что нам объясняли на лабе	Применил знания на практике	Удивился с того, насколько хороший алгоритм получился

10. Замечания автора

Замечаний нет! Умение применять битовые операции наверняка ещё пригодится, потому что это очень эффективно.

11. Выводы

В общем, работа понравилась, не приходилось раньше работать с битовыми операциями. Причём, приём с тем, что мы запикиваем целый алфавит в одно число, мне очень понравился: что по памяти, что по скорости.

Анекдот (его придумал мой сосед, гений, добряк, филантроп, Александр Шляхтуров из 910 комнаты):

Что придумал Коши, когда лёг?

Кошелек

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

Подпись студента _____