

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

Студент группы **М8О-107Б-20** Чекменев Вячеслав Алексеевич, № по списку 25

Контакты e-mail: chekmenev031@gmail.com, telegram: @suraba03

Работа выполнена: «13» декабря 2020 г.

Преподаватель: каф. 806 Найденов Иван Евгеньевич

Отчет сдан « » _____ 20 ____ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Множества.

2. **Цель работы:** Составить программу проверки характеристик введенных последовательностей слов и печати развернутого ответа.

3. **Задание ():** вариант 19

4. **Оборудование** (студента):

Процессор *Intel Core i5-8265U* с ОП 7851 Мб, НМД 256 Гб. Монитор 1920x1080

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

Операционная система семейства UNIX: linux, наименование: manjaro, версия: 20.1 Mikah

интерпретатор команд: bash, версия: 5.0.18.

текстовый редактор: code - oss

Утилиты операционной системы: –

Прикладные системы и программы: GNOME terminal

Местонахождение и имена файлов программ и данных –

6. **Идея, метод, алгоритм**

идти слева направо циклом вайл, увидев гласную, записать ее в множество, если до конца слова не появится другая главная, ввести yes, иначе продолжить итерации, обнулить множество.

7. **Сценарий выполнения работы**

Сценарий:

1) написать функции: ...,

2) читать посимвольно циклом вайл пока не eof .

3) проверять количество символов в множестве в каждой итерации, если к концу слова будет не 1 символ, то продолжить те же действия, иначе выйти из цикла.

4) за циклом поставить условие, если check_out == 1, ввести yes, иначе no

Тесты:

1) input: gfhgfhg yhhfggd

Output: no

2) input: sapiens homo

Output: yes

3) input: sapiens homo habilis

Output: yes

4) input: affirno

Output: no

5) input: kouurb jhhhzb

Output: no

8. Распечатка протокола

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

int is_split_simb(int c)
{
    return c == ' ' || c == '\t' || c == ',' || c == '\n';
}

uint32_t char_to_set(char c)
{
    if (c < 'a' || c > 'z') {
        return 0;
    } else {
        return 1u << (c - 'a');
    }
}

int main(void)
{
    uint32_t VOWELS = (1u << ('a' - 'a') | 1u << ('e' - 'a') | 1u << ('i' - 'a') | 1u << ('o' - 'a') | 1u << ('u' - 'a'));
    int c, check_out = 0;
    uint32_t letters_set = 0;

    while ((c = getchar()) != EOF) {
        if (!(is_split_simb(c))) {
            if (((1u << (c - 'a')) & VOWELS) != 0) {
                letters_set = letters_set | char_to_set(c);
                if (((letters_set & ~(1u << ('a' - 'a')) == 0)) {
                    check_out = 1;
                } else if (((letters_set & ~(1u << ('e' - 'a')) == 0)) {
                    check_out = 1;
                } else if (((letters_set & ~(1u << ('i' - 'a')) == 0)) {
                    check_out = 1;
                } else if (((letters_set & ~(1u << ('o' - 'a')) == 0)) {
                    check_out = 1;
                } else if (((letters_set & ~(1u << ('u' - 'a')) == 0)) {
                    check_out = 1;
                } else {
                    check_out = 0;
                }
            }
        } else {
            if (check_out == 0) {
                letters_set = 0;
            } else {
                break;
            }
        }
    }
    if (check_out == 1) {
```

```
    printf("Yes\n");  
} else {  
    printf("No\n");  
}  
return 0;  
}
```

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
1 3	дом	17 дека бря	19:00	Все получилось сразу	-	

10. Замечания автора по существу работы

Вроде все хорошо.

11. Выводы

Работа оказалась интресной, долго сокращал программу, поэтому сразу не получалось.

Подпись студента: