

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

Студент группы М80-109Б-22 Серый Никита Олегович, № по списку 16

Контакты:

email — [nikita.seryj@mail.ru](mailto:nikita.seryj@mail.ru)

telegram - @hukumkas

Работа выполнена: «31» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «6» декабря 2022 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

## 1. Тема: Множества

2. **Цель работы:** Составить программу проверки характеристик введенных последовательностей слов и печати развернутого ответа в соответствии с вариантом задания.

3. **Задание (вариант № 13):**  
Есть ли гласная, не входящая ни в одно слово?

4. **Оборудование (студента):**

Процессор *AMD Ryzen 5 5500U @ 6x 2.1GH* с ОП *15360 Мб*, НМД (?) *512 Гб*. Монитор *1920x1080*

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

Операционная система семейства: *linux*, наименование: *mint*, версия *21 Cinnamon*

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)

1. Создаём функцию `is_separator`, определяющую, является ли входной символ разделителем.

2. Функция `letter_to_int` проверяет входной символ на то, является ли он буквой..

3. Функция `letter_to_set` вычисляет значение `letter_to_int` по входному значению и возвращает значение в виде множества.

4. В `check_letters` составляем множество гласных из объединения множеств по одной гласных.

5. Если не встречаем EOF, объединяем `set_all_letters`, который равен нулю, со считанным символом.

6. Если все же встретили EOF, то смотрим на пересечение с множеством всех гласных. Тогда и выносится строка вердикта.

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

Входные данные	Выходные данные	Описание тестируемого случая
qwertyuiop[[[poiu uytrew	Not all vowel letters are in the line.	Действительно, не все гласные в строке.
qwertyuiop[asdfg hijkl;xcvbnm,,	All vowel letters are in the line.	А здесь все гласные в строке.

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

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

```

#include <stdbool.h>
#include <assert.h>

typedef unsigned long long set;

bool is_separator(int sym) {
    return (sym == ' ' || sym == '\t' || sym == '\n');
}

void test_is_separator() {
    assert(is_separator(' ') == 1);
    assert(is_separator('\t') == 1);
    assert(is_separator('\n') == 1);
}

int letter_to_int(int sym) {
    if (sym >= 'a' && sym <= 'z')
        return sym - 'a';
    if (sym >= 'A' && sym <= 'Z')
        return sym - 'A';
    else
        return -1;
}

set letter_to_set(int sym) {
    int ind = letter_to_int(sym);
    return (1U) << ind;
}

void check_letters() {
    int c;
    set all_letters = 0;
    set vowel_letters = letter_to_set('A') | letter_to_set('E') | letter_to_set('I') | letter_to_set('O') | letter_to_set('U') |
letter_to_set('Y');
    while (1) {
        c = getchar();
        if (is_separator(c)) {
            continue;
        } else if (c == EOF) {
            if (((all_letters & vowel_letters) == vowel_letters)) {
                printf("\nAll vowel letters are in the line. \n");
            } else {
                printf("\nNot all vowel letters are in the line.\n");
            }
            break;
        } else {
            all_letters = all_letters | letter_to_set(c);
        }
    }
}

int main() {
    test_is_separator();
    check_letters();
    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

#### 10. Замечания автора по существу работы

#### 11. Выводы

Интересная работа. Узнал, что в СИ работа с множествами реализуется за счёт битовых операций языка. Узнал также про битовую маску. Мне понравилась работа.

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

Подпись студента

---