

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

Студент группы Алапанова Эльза Халилевна, № по списку 3

Контакты e-mail : alapanowa02@yandex.com

Работа выполнена: «16» января 2021г.

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 №20):** Есть ли слово, содержащее одну согласную, возможно несколько раз.
4. **Оборудование** (студенческое)

Процессор Intel® Core™ i5-10210 @ 1.60 GHz с ОП 8192 Мб, НМД 512 Уб. Монитор 1920 x 1080

Программное обеспечение (студенческое):

Операционная система семейства Windows, наименование Windows 10 домашняя версия
интерпретатор команд _____ версия _____.

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

Прикладные системы и программы Sublime Text

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

На вход подаётся произвольный набор английских слов, разделённых пробелами, запятыми, знаками табуляции и границами строк. Букву 'у' для простоты считать согласной (гласными или согласными бывают звуки, но не буквы).

Необходимо проверить выполнение следующего условия: есть ли слово, содержащее одну согласную, возможно несколько раз? При решении задачи необходимо реализовать математическую абстракцию множества (сам тип, полное/пустое множество, пересечение, симметрическая разность и т.д.). Для реализации множества необходимо использовать битовые операции, но при этом запрещается нарушать принципы абстракции и инкапсуляции: пользовательский код должен зависеть только от интерфейса, но не от его реализации.

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

План работы:

Слова во входной последовательности имеют произвольную длину и состоят только из строчных и прописных букв английского алфавита. Слова могут быть обрамлены любым количеством пробелов, запятых, знаков табуляции и переводов строк. Входной текст закодирован в произвольной кодировке, содержащей ASCII в качестве подмножества и допускающей ASCII-интерпретацию.

Вывести Yes, если результат проверки положительный, и No в противном случае. Как всегда, вывод должен завершаться Unix-переводом строки.

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

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

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

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

```
int rubbish(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');
    }
}
```

```
char to_lower(char c)
```

```
{
    if (c >= 'A' && c <= 'Z') {
        c = c - 'A' + 'a';
    }
    return c;
}
```

```
int main(void)
```

```
{
    uint32_t VOWELS = (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 << ('y' - 'a') | 1u << ('z' - 'a'));
    int c, check = 0;
    uint32_t letters = 0;
```

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

```
    c = to_lower(c);
```

```
    if (!(rubbish(c))) {
```

```
        if (((1u << (c - 'a')) & VOWELS) != 0) {
```

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

```
            if (((letters & ~(1u << ('b' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('c' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('d' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('f' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('g' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('h' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('k' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('l' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('m' - 'a')) == 0)) {
```

```
                check = 1;
```

```
            } else if (((letters & ~(1u << ('n' - 'a')) == 0)) {
```

```
                check = 1;
```

```

    } else if (((letters & ~(1u << ('p' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('q' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('r' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('s' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('t' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('v' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('w' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('x' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('y' - 'a'))) == 0)) {
        check = 1;
    } else if (((letters & ~(1u << ('z' - 'a'))) == 0)) {
        check = 1;
    } else {
        check = 0;
    }
}
} else {
    if (check == 0) {
        letters = 0;
    } else {
        break;
    }
}
}
if (check == 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						
2						

10. Замечания автора по существу работы : замечаний нет.

11. Выводы : очень трудно дается программирование на Си..

Недочёты при выполнении задания могут быть устранены следующим образом: _____
 _____ Подпись студента _____