



Отчёт по лабораторной работе № 13 по курсу Выч. Системы

Студент группы М8О-106Б-21 Орусский В.Р., № по списку 15

Контакты www, e-mail, icq, skype slava111103@gmail.com

Работа выполнена: « » 201 г.

Преподаватель: Ст. Преп. каф.806 Двбинин А.В.

Входной контроль знаний с оценкой

Отчёт сдан « » 201 г., итоговая оценка

Подпись преподавателя

1. Тема: Множества

2. Цель работы: Изучить множества и их реализацию в коде. Необходимо в соответствии с заданием обработать строку или строки, поданные на вход, между словами могут использоваться любые разделители. Используя множества проверить характеристики данных в строках слов. Для русского алфавита необходимо учитывать представление кириллицы в Unicode

3. Задание (вариант №19): Есть ли слово, содержащее одну гласную (возможно более одного раза)

4. Оборудование(лабораторное):

ЭВМ , процессор , имя узла сети с ОП Мб, НМД Мб. Терминал адрес . Принтер Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор с ОП Мб, НМД Мб. Монитор Другие устройства

5. Программное обеспечение(лабораторное):

Операционная система семейства , наименование версия интерпретатор команд версия Система программирования версия Редактор текстов версия Утилиты операционной системы

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства , наименование версия интерпретатор команд версия Система программирования версия Редактор текстов версия 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)

Алгоритм:

- 1) Берём число и находим его номер в Unicode (с помощью его представления в utf-8 через какое-то отведённое кол-во байт).
- 2) Распределяем полученные буквы в множество по номеру в алфавите.
- 3) Создаём переменную, которая будет хранить позиции гласных в обоих алфавитах.
- 4) Заполняем множество букв текущего слова, когда слово закончится применяем побитовое умножение между множеством слова и множеством гласных. Если получим число, являющееся степенью двойки, то это означает, что в множестве одно единственное пересечение (в одной гласной)

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

```
int letter_in_utf8() {
    unsigned int sym_code = getchar();
    if (sym_code == EOF) {
        return -1;
    }
    if (sym_code <= 127) {
        return sym_code;
    } else {
        unsigned int count_of_bytes = sym_code - 192;
        int k = 2;
        if (count_of_bytes & 100000) {
            k++;
            count_of_bytes -= 32;
            if (count_of_bytes & 10000) {
                k++;
            }
        }
        count_of_bytes = sym_code;
        count_of_bytes = count_of_bytes >> (8 - k + 1);
        unsigned int leftover_bytes;
        leftover_bytes = sym_code << (32 - (8 - (k + 1)));
        leftover_bytes = leftover_bytes >> (32 - (8 - (k + 1)));
        while (count_of_bytes) {
            sym_code = getchar();
            sym_code = sym_code << 26;
            sym_code = sym_code >> 26;
            leftover_bytes = leftover_bytes << 6;
            leftover_bytes = leftover_bytes + sym_code;
            count_of_bytes = count_of_bytes >> 1;
        }
        return leftover_bytes;
    }
}
```

```
int ord_from_alphabet(int alpha) {
    int big_A = 1040, big_E = 1045, big_YO = 1026, big_YA = 1071, little_a = 1072, little_e = 1077, little_yo = 1106,
        little_ye = 1101;
    if (alpha > big_A && alpha <= big_E) {
        return alpha - big_A;
    }
    if (alpha > big_E && alpha <= big_YA) {
        return alpha - big_A + 1;
    }
    if (alpha > little_a && alpha <= little_e) {
        return alpha - little_a;
    }
    if (alpha > little_e && alpha <= little_yo) {
        return alpha - little_a + 1;
    }
    if (alpha == big_YO || alpha == little_yo) {
        return 6;
    }
    if (alpha > 'A' && alpha <= 'Z') {
        return 33 + alpha - 'A';
    }
    if (alpha > 'a' && alpha <= 'z') {
        return 33 + alpha - 'a';
    }
    return -1;
}
```

В качестве множеств мы используем тип long long, он занимает 64 бита, что позволяет разместить порядка 64 уникальных значений.

Выше представлены функции представления числа в Unicode и определение номера буквы в алфавите. Далее, мы просто считываем слово и каждую букву в нём направляем во множество word. Заранее создаём множество со всеми гласными двух алфавитов. Когда слово заканчивается применяем побитовое перемножение, везде где было 1 и 1, останется единица, то есть в местах пересечения этих множеств находится гласная, нам важно, что находилась только одна. Это возможно тогда и только тогда, когда полученное число является степенью двойки (двоичное представление числа, если число имеет одну единицу и все нули, то это степень двойки).

P.S. Во множества буквы распределяются на позиции под номерами в алфавите (буква а на первой позиции в множестве, буква г - 4 и так далее.

Тест: ввод: "альфа bravo янки mag"; Вывод: count of words:2

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

Допущен к выполнению работы. Подпись преподавателя _____

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

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

В данной лабораторной работе я научился переводить буквы в их десятичный номер в Unicode, использовать побитовый сдвиг, побитовое умножение. Данные знания будут очень полезны, поскольку это новое представление множеств для меня, без использования встроенных библиотек, аналог множества был реализован собственными руками.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента

