



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФГБОУ ВПО Брянский государственный технический

УТВЕРЖДАЮ

Ректор университета

_____ А.В. Лагереv

« ____ » _____ 2012 г.

**СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ
ПРИНЦИПЫ РАБОТЫ С ТЕКСТОВЫМИ ФАЙЛАМИ**

**Методические указания
к выполнению лабораторной работы №1
для студентов очной формы обучения по направлению
230100 «Информатика и вычислительная техника» (профиль
«Программное обеспечение вычислительной техники и
автоматизированных систем»)**

Брянск 2012

УДК 004.422.63

Структуры и алгоритмы обработки данных. Принципы работы с текстовыми файлами [Текст] + [Электронный ресурс]: методические указания к выполнению лабораторной работы №1 очной формы обучения по направлению 230100 «Информатика и вычислительная техника» (профиль «Программное обеспечение вычислительной техники и автоматизированных систем»). – Брянск: БГТУ, 2012. – 16 с.

Разработали:
В.К. Гулаков
канд. техн. наук, проф.
А.О. Трубаков
канд. техн. наук, доц.
С.Н.Зимин
ст.преп.

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ (протокол №1 от 31.08.12)

1. ЦЕЛЬ РАБОТЫ

Цель работы – научиться эффективно обрабатывать информацию, находящуюся во внешней памяти (в файлах). Для этого студент должен

- изучить способы работы с текстовыми файлами;
- освоить основные алгоритмы обработки символьной информации.

В результате выполнения работы студент должен знать функции чтения и записи в файл и работы со строками.

Продолжительность работы - 2 часа.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Общие принципы организации

2.2. Основные функции работы с файлами

2.3. Символьная обработка файла

2.4. Блочная работа с файлами

2.5. Работа с текстовой информацией на языке C++

Стандартные текстовые функции.

Класс string.

Расширение boost (спрайты и регулярные выражения).

3. ЗАДАНИЕ

Разработать схему, описание алгоритма и программу для обработки символьной информации в соответствии с вариантом задания.

Исходный текст разместить в файле. Из результатов обработки этого текста сформировать результат и ввести его в файл результатов. Исходные тексты формировать самостоятельно.

Индивидуальные варианты заданий. В каждом варианте задания исходным является текст, состоящий из нескольких абзацев. Слова предложений разделяются пробелами и знаками препинания; предложения – точкой или знаками '?' и '!'; абзац – символом перевода каретки.

1. Имеется файл с текстом на русском языке. Дать варианты переноса всех слов. Перенос возможен по следующим правилам:

- переносятся либо остаются в конце строки не менее двух символов;
- невозможен перенос перед буквами 'ь' и 'ъ';
- слово должно иметь не менее двух слогов;
- в оставшейся и переносимой частях слова должны быть гласные буквы.

2. Текстовый файл содержит последовательность целых чисел, разделенных произвольным количеством других символов (текст, в котором встречаются числа). Требуется найти сумму чисел, а так же сформировать выходной файл, в котором все числа будут заменены на '###'.

3. Строки текстового файла содержат фамилии студентов в произвольном порядке. Переносы в файле разрешены. Определить порядковый номер байта, начиная с которого располагается первая по алфавиту фамилия. Заменить данную фамилию в исходном файле символами '*', не переписывая этого файла.

4. Дан текстовый файл и ключевая фраза для поиска. Выдать на экран 5 строк файла, включающей заданный образец (две строки до найденного образца и две – после). Выделить образец с двух сторон символами '*'.

5. Дан текстовый файл и два слова. Переносы слов допускаются. Получить новый файл, в котором все вхождения первого слова заменены на второе слово.

6. В некоторых строках текстового файла имеются выражения, состоящие из двух целых чисел, разделенных знаком арифметической операции ('+', '-', '*', '/'). В строке перед выражением и после него могут

находиться произвольные символы. Требуется заменить все выражения на результат их расчёта.

7. Имеется два текстовых файла. В первом из них содержится некоторое описание. Переносы слов допускаются. Второй файл содержит список слов, не подлежащих разглашению. Требуется переписать первый файл, заменив каждое из подобных слов символом '*'.

8. В файле записан текст стихотворения. Форматировать текст так, чтобы каждый куплет из 4 строк следовал с одной и той же позиции, начинался с прописной буквы и был сдвинут относительно предыдущего куплета на 5 позиций вправо или влево поочередно.

9. Некоторый текст состоит из нескольких частей, записанных в отдельных файлах. Имена этих файлов и заголовки для них указаны в отдельном файле. Создать файл с полным текстом. Заголовок каждой части должен содержаться в центре первой строки.

10. Имеется текстовый файл. Требуется проверить, является ли первая буква каждого предложения прописной. Так же проверить, есть ли в файле разделители слов, состоящие из нескольких подряд идущих пробелов. Исправить обнаруженные ошибки.

11. Алфавит некоторого языка программирования включает латинские буквы, цифры, знаки пунктуации (.,;:!?) и языки арифметических операций (+-* /). Программа, написанная на этом языке, содержится в текстовом файле. Проверить допустимость текста программы. Выдать сообщение с указанием места ошибочных символов.

12. Имеется текстовый файл, состоящий из нескольких разделов и подразделов. Разделы нумеруются одной цифрой (например: 1), подразделы несколькими цифрами, разделенными точкой. Сформировать выходной файл, в котором сначала будет идти оглавление текста (без номеров страниц, только заголовки), а затем сам текст. Других цифр, кроме номеров разделов, в тексте нет. Подзаголовки в оглавлении должны быть сдвинуты с помощью пробелов вправо.

13. Дан текстовый файл и одно слово. Найти в текстовом файле слова, отличающиеся от заданного на одну букву и заменить на заданное. Выделить их подсветкой.

14. Определить, сколько слов текста начинается и заканчивается на одну и ту же букву. Вывести эти слова.

15. Определить сколько слов текста имеют длину 1, 2, 3, ... 10 символов. Вывести все слова через запятую, длиннее 8 символов.

16. Дан символ. Определить номера позиций слов, в которых размещен данный символ. Вывести эти слова и номера позиций.

17. Дан символ (2 пробела, минус, звездочка или точка) и произвольный текст в файле. Сформировать выходной файл, в котором перечислены все слова из первого файла и одно слово от другого отделяется заданным символом.

18. Исключить из исходного текста части текста в круглых скобках вместе со скобками. Вложенных скобок в тексте нет. Оставшийся текст "сжать".

19. Даны N и I - номер строки и позиция в строке. Дана строка для вставки ее в текст. Вставить заданную строку в текст после I символа N-строки исходного текста.

20. Дано N. Выделить из текста каждый N-й символ каждого слова и сформировать из них строку.

21. В одном файле записан словарь (каждое слово начинается с новой строки и состоит только из букв). Максимальный размер словаря известен – не более 100 слов. Необходимо найти самое короткое слово из словаря, которого нет во входном тексте. Так же необходимо все слова из словаря в тексте обрмить звездочками (добавить в начало и в конец слова по одному символу '*').

22. В входном файле дан произвольный текст. Необходимо вывести следующую статистическую информацию:

- количество слов и символов в каждом предложении;
- среднее количество слов в предложении;
- среднее количество символов в предложении;
- среднее количество букв в слове.

23. В входном файле дан произвольный текст. Необходимо переписать этот текст так, что бы все слова в выходном файле были отображены зеркально (обратный порядок букв в словах). При этом знаки пунктуации и конца абзаца должны остаться на неизменных позициях.

24. Дан текст, состоящий из слов на русском и английском языках. Вывести в алфавитном порядке:

- все гласные буквы, на которые в тексте начинается только одно слово;
- все согласные буквы, на которые в тексте начинается более одного слова;
- по одному любому слову из текста для каждой буквы, которая удовлетворяет описанным критериям.

25. Имеется некоторый файл с текстом и файл содержащий отдельные слова (словарь). Максимальный размер словаря известен – не более 100 слов. Разработать программу, проверяющую правильность написания отдельных слов, а так же исправляющий одиночные ошибки (слова, в которых написан один символ неправильно). В случае отсутствия слова в словаре – выводить соответствующее сообщение.

26. Дано строка из 2-х символов. Подсчитать, сколько раз встречается в тексте эта подстрока. Вывести эти слова.

27. Дан символ. Удалить из текста каждый заданный символ и повторить каждый символ, не равный заданному.

28. Дано натуральное число n ($n \leq 1000$). Записать это число русскими словами (семнадцать, двести пятьдесят три и т.д.).

29. Определить в тексте количество попаданий каждой гласной и каждой согласной буквы. Упорядочить буквы по частоте их попаданий в тексте.

30. Дана строка из 2-х символов. Найти в тексте заданную подстроку и удалить ее из всех слов текста. Оставшийся текст "сжать".

Дополнительное задание повышенной сложности.

- Обработку файла сделать на основе блочной парадигмы. При этом программа должна работать при любом размере файла и любом размере абзаца.
- Обязательно учесть в программе, что между словами могут находиться несколько пробелов, в тексте могут встречаться пустые абзацы (несколько подряд идущих переводов строк), могут попадаться цифры и числа. В выходном файле лишние пробелы и переводы строк должны отсутствовать.
- Программа должна быть регистронезависимой, т.е. большая и маленькая буква считаются одной и той же (если обратного не предусмотрено индивидуальным заданием).
- Название входного и выходного файлов, а также все необходимые переменные, должны запрашиваться у пользователя.
- Необходимо уметь обосновать выбранные размеры переменных и временных буферов.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

В процессе выполнения лабораторной работы студенту необходимо выполнить следующие шаги.

1. Используя текстовый редактор, сформировать файл с исходными данными.
2. Написать программу, выполняющую над текстом указанное задание для своего варианта и выводящую сформированные результаты в новый файл.
3. Проверить корректность программы на нескольких тестах. В случае необходимости исправить найденные ошибки и недочеты.
4. Сформировать отчет и защитить лабораторную работу.

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет студента о проделанной работе должен содержать следующие части:

- задание на лабораторную работу;
- описание алгоритма решения задачи;
- описание основных блоков и функций;
- исходный текст программы;
- тестовые примеры (со скриншотами экранов);
- файлы тестовых примеров.

Отчет сдается в электронном виде.

6. ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЯ

В примере не ставилась задача обеспечения удобного интерфейса с пользователем. Для этих целей можно высвечивать информацию в выделенных на экранах окнах, в режиме "меню" спрашивать пользователя о его намерениях, организовать начальную заставку с информацией о программе, добавить элементы звукового сопровождения т.д. Пример реализован на языке программирования C++.

Задание.

Дан текстовый файл, в котором возможны переносы слов со строки на строку. Подсчитать общее число слов. Имя файла задать в командной строке.

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

//=====
// Главная функция программы
//=====
void main (int argc, char *argv[])
{ // Открытие исходного файла
  if(argc<2)
  { printf("Не указан исходный файл");
    return;
  }
  FILE *FileIn = fopen(argv[1], "r");
  if(FileIn == NULL)
  { printf("Невозможно открыть указанный файл");
```

```

        return;
    }

// обработка файла
int Kol=0; // Количество найденных слов
char Buf[512]; // Буфер для считывания символов
bool IsCurLetter; // Флаг - "текущий символ буква?"
bool IsCurPerenos; // Флаг - "текущий символ перенос?"
bool IsPrevLetter=false; // Флаг - "предыдущий символ буква?"
bool IsPrevPerenos=false; // Флаг - "предыдущий символ перенос?"
while(!feof(FileIn))
{
    int n = fread(Buf, sizeof(char), 512, FileIn);
    for(int i=0; i<n; i++)
    {
        // проверка, можно ли считать текущий символ буквой
        if((Buf[i]>='a' && Buf[i]<='z') ||
            (Buf[i]>='A' && Buf[i]<='Z') ||
            (Buf[i]>='а' && Buf[i]<='я') ||
            (Buf[i]>='А' && Buf[i]<='Я') ||
            (Buf[i]>='0' && Buf[i]<='9'))
            IsCurLetter = true;
        else
            IsCurLetter = false;
        // проверка, можно ли считать текущий символ переносом
        if((Buf[i]=='-' && IsPrevLetter) ||
            ((Buf[i]=='\r' || Buf[i]=='\n') && IsPrevPerenos))
            IsCurPerenos = true;
        else
            IsCurPerenos = false;
        // проверка, является ли концом слова
        if(!IsCurLetter && !IsCurPerenos &&
            (IsPrevLetter || IsPrevPerenos))
            Kol++;
        IsPrevLetter = IsCurLetter;
        IsPrevPerenos = IsCurPerenos;
    }
}
if(IsPrevLetter || IsPrevPerenos)
    Kol++;

// Закрывать файл и вывести результат на экран
fclose(FileIn);
printf("Количество слов: %d\n", Kol);
getch();
}

```

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. С помощью каких процедур можно открыть и закрыть сформированный текстовый файл?

2. С помощью каких процедур можно производить обмен данными между ОП и сформированным текстовым файлом?
3. Что из себя представляет кодировка ASCII?
4. Как найти код заданного символа?
5. Какие виды кодировок вы знаете?
6. Чем отличаются различные виды кодировок?
7. Какие операции можно выполнять над символами?
8. Какие операции можно выполнять над строками?
9. Как определяются символы и строки в различных алгоритмических языках?

8. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Трамбле Ж., Соренсон П. Ведение в структуры данных. - М.: Машиностроение, 1982. - 784 с.
2. Райли Д. Абстракция и структуры данных: Вводный курс: Пер. с англ. -М.: Мир, 1993.- 752 с., ил.
3. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста: Пер. с англ./Ричард Хэзфилд, Лоуренс Кирби и др. – К.: Издательство «ДиаСофт», 2001. – 736 с.