

Лабораторная работа 1: Перекодировка

Цель работы: изучить особенности работы с массивами и строками в C.

Описание:

Программа должна выполнять перекодировку текстового файла.

Аргументы программе передаются через командную строку:

lab1 <имя_входного_файла> <имя_выходного_файла> <выходная_кодировка>

где <выходная_кодировка>:

- 0 – UTF-8 без BOM;
- 1 – UTF-8 с BOM;
- 2 – UTF-16 Little Endian;
- 3 – UTF-16 Big Endian;
- 4 – UTF-32 Little Endian;
- 5 – UTF-32 Big Endian.

Входная кодировка: любая из перечисленных, должна определяться программой автоматически.

Неправильные UTF-8 байты кодировать/декодировать как символы из диапазона 0xDC80..0xDCFF. UTF-8 -> другая кодировка -> UTF-8 должно точно возвращать исходный файл (плюс/минус BOM).

Программа должна:

- 1) быть написана на C;
- 2) не использовать внешние библиотеки;
- 3) всегда корректно освобождать память и закрывать файлы;
- 4) обрабатывать ошибки: файл не открылся, не удалось выделить память – выдавать сообщение об ошибке и корректно завершаться с ненулевым кодом возврата (при отсутствии ошибок – завершаться с нулевым);
- 5) не писать в консоль ничего лишнего, кроме сообщений об ошибках и краткой справки по использованию (при запуске с неправильным числом аргументов).

Лабораторная работа 2: Сортировка

Цель работы: изучить особенности работы со строками, структурами и рекурсией в C.

Описание:

Программа должна сортировать (регистрозависимо, однобайтовая кодировка символов) строки файла в формате:

<фамилия><пробел><имя><пробел><отчество><пробел><телефон>

где:

<фамилия>, <имя>, <отчество> – строки не длиннее 20 символов, не содержат пробелов, табуляции, переводов строк или ноль-символов;
<телефон> – целое неотрицательное число, меньшее 10^{11} .

В порядке фамилия->(если совпадает)имя->отчество->телефон.

Каждый элемент строки должен храниться отдельным полем структуры. Порядок сортировки должно быть возможно изменить (например, первым учитывать имя) незначительными изменениями в программе.

Аргументы программе передаются через командную строку:

lab2 <имя_входного_файла> <имя_выходного_файла>

Программа должна:

- 1) быть написана на C;
- 2) не использовать внешние библиотеки;
- 3) метод сортировки: самостоятельно реализованный QuickSort (детерминированный, не должен падать при любых входных данных);
- 4) всегда корректно освобождать память и закрывать файлы;
- 5) обрабатывать ошибки: файл не открылся, не удалось выделить память – выдавать сообщение об ошибке и корректно завершаться с ненулевым кодом возврата (при отсутствии ошибок – завершаться с нулевым);
- 6) не писать в консоль ничего лишнего, кроме сообщений об ошибках и краткой справки по использованию (при запуске с неправильным числом аргументов).

Лабораторная работа 3: Решение системы уравнений

Цель работы: изучить особенности работы с числами с плавающей точкой в C.

Описание:

Программа должна находить решение системы линейных уравнений.

Входной файл в первой строке содержит одно число: размер системы уравнений N, после чего идут N строк по N+1 числу, которые соответствуют коэффициентам при N переменных и свободному члену.

Выходной файл должен содержать:

- найденные значения переменных в формате одно число на каждой строке если решение единственно;
- только фразу “many solutions” если решение не единственно;
- только фразу “no solution” при отсутствии решений.

Пример входных данных:

```
2
0.5 3 4
0 2 5
```

Пример выходных данных:

```
-7
2.5
```

Аргументы программе передаются через командную строку:

lab3 <имя_входного_файла> <имя_выходного_файла>

Программа должна:

- 1) быть написана на C;
- 2) не использовать внешние библиотеки;
- 3) всегда корректно освобождать память и закрывать файлы;
- 4) обрабатывать ошибки: файл не открылся, не удалось выделить память – выдавать сообщение об ошибке и корректно завершаться с ненулевым кодом возврата (при отсутствии ошибок – завершаться с нулевым);
- 5) не писать в консоль ничего лишнего, кроме сообщений об ошибках и краткой справки по использованию (при запуске с неправильным числом аргументов).

Лабораторная работа 4: Длинная арифметика

Цель работы: изучить особенности работы со структурами в C/C++.

Описание:

Программа должна выполнять арифметические операции над целыми числами произвольной точности.

Входной файл в первой строке содержит первое число, вторая строка содержит знак операции ('+', '-', '*', '/', '%') – остаток от деления, '#' – корень квадратный, '<', '<=', '>', '>=', '==', '!='), третья строка (за исключением операции квадратного корня) содержит второе число.

Выходной файл должен содержать одно целое число: результат операции (для операций сравнения – 0 или 1; для квадратного корня – целую часть результата; при взятии корня из отрицательного числа или делении на ноль – NaN; -0 не может быть результатом).

Аргументы программе передаются через командную строку:

lab4 <имя_входного_файла> <имя_выходного_файла>

Программа должна:

- 1) быть написана на C или C++;

- 2) не использовать внешние библиотеки;
- 3) всегда корректно освобождать память и закрывать файлы;
- 4) обрабатывать ошибки: файл не открылся, не удалось выделить память – выдавать сообщение об ошибке и корректно завершаться с ненулевым кодом возврата (при отсутствии ошибок – завершаться с нулевым);
- 5) не писать в консоль ничего лишнего, кроме сообщений об ошибках и краткой справки по использованию (при запуске с неправильным числом аргументов);
- 6) умножение, деление и квадратный корень должны работать с адекватной скоростью (то есть требуется алгоритм уровня “в столбик”, а не “умножение на n путём сложения n раз”).

Рекомендации по решению:

- 1) Каждое число представлять структурой/классом, где отдельно хранится знак числа и отдельно разряды числа, при этом разряды рекомендуется хранить массивом, начиная с младшего (`digits[0]` соответствует самой младшей цифре числа).
- 2) Реализовать вспомогательные функции/методы: сложение, вычитание, универсальное сравнение (возвращает -1, 0, 1), которые будут использоваться в функциях/методах ‘операция +’, ‘операция <’,...
- 3) При реализации на C++: методы, реализующие операции, сделать перегрузками этих операций для данного класса; сделать конструктор из строки символов.