Пермский национальный исследовательский политехнический университет

Кафедра «Автоматика и телемеханика»

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ

(УКАЗАТЕЛИ, СТРОКИ, ТЕКСТОВЫЕ ФАЙЛЫ)

Методические указания к индивидуальной работе №2

по дисциплине «Программирование и основы алгоритмизации»

Пермь 2015

Введение

Цель работы: получение навыков составления на языке С программ, в которых используются указатели, строки и текстовые файлы.

Студент самостоятельно выполняет задание для данной индивидуальной работы согласно своему варианту. Затем должен быть оформлен отчет по выполненной работе согласно требованиям к оформлению.

Зачет по данной работе ставится по итогам ее защиты. В ходе защиты студент должен: продемонстрировать выполнение разработанной программы; предоставить правильно оформленный отчет по индивидуальной работе; ответить на вопросы преподавателя, касающиеся особенностей разработанной программы.

Задание

Требуется разработать программу, которая преобразует любой текстовый файл в другой текстовый файл по правилу, определяемому на основе своего варианта согласно табл. 1. Указанное правило применяется для каждой строки исходного текстового файла. Номер варианта задания выдается преподавателем.

Дополнительные требования:

1) в ходе выполнения программы пользователь должен иметь возможность ввести имя исходного текстового файла, а также имя получаемого текстового файла;

2) преобразование строки должно быть реализовано в виде функции, при этом один аргумент функции – это указатель на исходную строку, второй аргумент – это указатель на получаемую строку.

Рекомендации

Для начала можно составить программу, которая преобразует строку, введенную с помощью клавиатуры, согласно правилу преобразования строки. На данном этапе можно не обращать внимания на требования вида «в каждой второй строке», «вместо каждой третьей строки», если такие требования есть в варианте задания.

Правило преобразования строки сначала можно не оформлять в виде функции, а отладить программу в виде одной функции main. После того, как все будет работать правильно, можно реализовать указанное правило в виде функции, проверив ее выполнение.

Затем рекомендуется перейти к работе с текстовыми файлами.

Текстовые файлы лучше создавать в папке текущего проекта, то есть там, где находится файл с текстом разрабатываемой программы.

Тем, кто затрудняется с созданием текстового файла, можно поступать следующим образом. 1) В «Проводнике» или через «Мой компьютер» надо перейти в папку проекта (где находится файл с текстом разрабатываемой программы). 2) Нажать *правую* кнопку мыши на свободное поле в этой папке, например, ниже последнего файла в этой папке. Появится контекстное меню, в

котором надо выбрать пункт «Создать–>Текстовый документ». Затем ввести имя файла и нажать Enter. При этом лучше, чтобы имя файла заканчивалось на «.txt». Например, можно ввести имя «input.txt». Если потом надо будет поменять имя, то тогда достаточно нажать правую кнопку мыши на имени этого файла и из контекстного меня выбрать «Переименовать». 3) Надо изменить содержимое файла. Для этого можно дважды кликнуть мышью на имени файла. Откроется текстовый редактор, где можно ввести необходимый текст (последовательность строк). Затем надо сохранить файл, для этого в редакторе выбрать меню «Файл–>Сохранить». После этого редактор можно закрыть. Требуемый текстовый файл будет создан.

Ранее созданную программу, которая выполняет преобразование строки, можно модифицировать следующим образом. Предполагается, что преобразование строки реализовано в виде функции. Тогда функцию main можно изменить так, чтобы программа просто копировала (по строкам) один текстовый файл в другой. Потом останется модифицировать функцию main так, чтобы перед копированием очередной строки в выходной файл выполнялся (при необходимости) вызов функции, которая была ранее разработана для преобразования строки. Таким образом, строка из исходного файла будет преобразована согласно заданию и выведена в получаемый файл.

Таблица 1.

Правила преобразования текстового файла, соответствующие вариантам.

Номер варианта	Правило преобразования текстового файла (применяется для каждой строки исходного текстового файла)
11.	В каждой второй строке надо удалить третье слово, если такое есть, при этом количество пробелов в строке не должно измениться. Остальные строки не меняются.
12.	В каждой строке каждое слово заключается в круглые скобки, если данная строка начинается с цифры, при этом количество пробелов в строке не должно измениться.
13.	В каждой строке удаляются слова, которые состоят из одного или двух символов, при этом количество пробелов в строке не должно измениться.
14.	Если в строке четное количество слов, то удаляется каждое второе слово, при этом количество пробелов в строке не должно измениться.
15.	Если в строке есть хотя бы одно слово, первый и последний символ которого равны, то тогда поставить восклицательный знак в начало этой строки, сдвинув исходную строку на один символ вправо.
16.	В каждой строке каждое слово, состоящее более, чем из 8 символов, сократить до первых 8 символов этого слова, при этом количество пробелов в строке не должно измениться.
17.	В конец каждой строки через пробел добавляется число, которое равно количеству символов в самом длинном слове данной строки.
18.	Вместо каждой третьей строки оставить только наиболее длинное слово этой строки, если же таких слов несколько, то оставить только первое из них. Остальные строки не меняются.
19.	Если строка состоит из слов равной длины, то тогда поставить восклицательный знак в начало этой строки, сдвинув исходную строку на один символ вправо.

Требования к оформлению отчета

Отчет должен содержать:

– титульный лист;

– цель работы;

– задание согласно своему варианту с указанием варианта (предваряется заголовком «Задание»);

– исходный текст разработанной программы (предваряется соответствующим заголовком);

– пример выполнения программы в виде текста, выводимого на экран при выполнении программы, с указанием содержимого текстовых файлов.