Лабораторная работа 4.

Черновик 0.8

Целью работы является знакомство студентов со строками и их обработкой.

Студенты должны научиться

1. описывать строки;
2. обрабатывать строки с помощью функций из стандартной библиотеки (см. string.h) и без них;
3. передавать строки в функции;
4. создавать многофайловые проекты.

Общее задание

1. Исходный код лабораторной работы располагается в ветке lab_04. В этой ветке создается папка lab_04_1, в которой располагается решение первой задачи, и папки lab_04_X_Y – для исходного кода второй и третей задач (X - номер варианта, Y – номер задачи).

2. Исходный код должен соответствовать правилам оформления исходного кода.

3. Для каждой задачи создается отдельный проект в QT Creator. Для каждого проекта должно быть два варианта сборки: Debug (с отладочной информацией) и Release (без отладочной информации).
   Замечание
   По согласованию с преподавателем, проводящим практические занятия, вы можете использовать другую среду разработки (например, Microsoft Visual Studio Code), но два варианта сборки проекта нужно предусмотреть в любом случае.

4. Созданный проект обязательно должен быть многофайловым.

5. Для второй задачи подготавливаются тестовые данные, которые демонстрируют правильность ее работы. Входные данные должны располагаться в файлах in_z.txt, выходные out_z.txt, где z – номер тестового случая. Тестовые данные готовятся и помещаются под версионный контроль еще до того, как появится реализация задачи.

6. При компиляции программы необходимо использовать ключ “-Wvla”.

7. В условии каждой задачи указано какие действия можно выполнить с помощью стандартные функции для обработки строк, а какие действия реализуются самостоятельно.

8. Для первой задачи функция main содержит тесты, в которых сравнивается поведение реализованных функций с поведением стандартных функций.

9. Реализовав очередную задачу и проверив правильность ее работы, оцените полноту подготовленных тестовых данных на основе процента покрытия кода этими данными. Добейтесь 100% покрытия кода тестовыми данными. Если это невозможно, необходимо это обосновать.

10. Варианты во всех задачах распределяются преподавателем.

Индивидуальное задание

Задача 1

Самостоятельно реализовать указанные строковые функции.

0. strpbrk
1. strspn
2. strcspn
3. strchr
4. strrchr

Имена функций, которые реализуются самостоятельно, начинаются с префикса “my_” (если нужно реализовать функцию strpbrk, в программе она должна называться my_strpbrk).

Задача 2

Написать программу, которая запрашивает у пользователя одну или две строки, разбивает строку (или строки) на слова и выполняет обработку этих слов. Разбиение строки на слова реализуется самостоятельно (использовать для выделения слов функции scanf, sscanf или strtok нельзя).

В результате разбора строки должен быть сформирован массив слов (один или несколько). После чего выполняется обработка одного или нескольких массивов слов.

Длина строки не превышает 256 символов, длина слова - 16-ти символов. Слова разделяются одним или несколькими пробелами и знаками пунктуации (“,”, “;”, “:”, “-”, “.”, “!”, “?”).

Результаты выводятся обязательно после фразы "Result:" (кавычки не нужны). Слова (или слова и числа) разделяются одним пробелом.

В случае если задача решение не может быть получено, на экран ничего не выводится, возвращается код ошибки. Обратите внимание на ограничения по длине строке и по длине слова.

0. Ввести одну строку. Для каждого слова подсчитать количество его встреч в исходной строке. Программа должна вывести пары: “слово” “количество встреч” (кавычки не нужны). Каждая пара выводится на отдельной строке. Слова выводятся в том порядке, в котором они встретились в исходной строке.
1. Ввести одну строку. Составить массив из слов исходной строки (каждое слово должно входить в массив только один раз). Упорядочить этот массив в лексикографическом порядке. Слова из упорядоченного массива вывести на экран, разделив одним пробелом.
2. Ввести две строки. Для каждого слова из первой строки (повторяющиеся в этой же строке слова не обрабатываются) определить входит ли оно во вторую строку. Программа должна вывести пары: “слово” “yes/no” (кавычки не нужны). Каждая пара выводится на отдельной строке. Слова выводятся в том порядке, в котором они встретились в первой строке.
3. Ввести две строки. Напечатать слова, которые встречаются в двух строках только один раз (т.е. один раз либо в первой, либо во второй). Сначала выводятся слова из первой строки (в том порядке, в котором они встретились в этой строке), затем слова из второй строки.

Задача 3

Для решения этой задачи нужно использовать функции стандартной библиотеки для обработки строк.

Написать программу, которая запрашивает у пользователя строку, разбивает строку на слова. В результате разбора строки формируется массив слов.

Максимальные длины строки и слова, разделители слов такие же как в задаче 2.

Из слов, отличных от последнего, составляется новая строка, в которой слова разделяются одним пробелом. Слова в результирующую строку помещаются в обратном порядке. После последнего слова в результирующей строке пробел не добавляется.

Прежде, чем очередное слово помещается в результирующую строку, оно подвергается указанному преобразованию.

Если результирующая строка не пустая, она выводится на экран с помощью вызова функции printf следующим образом

```
printf("Result: %s\n", new_str);
```

Преобразование слова

0. Удалить из слова все последующие вхождения первой буквы.
1. Оставить в слове только первые вхождения каждой буквы.