

Лабораторная работа 6.

Черновик 0.55

Целью работы является знакомство студентов со строками и их обработкой.

Студенты должны научиться описывать строки, обрабатывать их самостоятельно и с помощью функций из стандартной библиотеки (см. `string.h`), передавать строки в функции.

1. Общее задание

1. Исходный код лабораторной работы располагается в ветке `lab_06`. В этой ветке создается папка `lab_06_1`, в которой располагается решение первой задачи, и папка `lab_06_X_2` – для исходного кода второй задачи (X - номер варианта).
2. В условии каждой задачи указано какие действия можно выполнить с помощью стандартные функции для обработки строк, а какие действия реализуются самостоятельно.
3. Исходный код должен соответствовать правилам оформления исходного кода.
4. Для первой задачи функция `main` содержит тесты, в которых сравнивается поведение реализованных функций с поведением стандартных функций.
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. Варианты в обоих задачах распределяются преподавателем.

2. Индивидуальное задание

Задача 1

Самостоятельно реализовать указанные строковые функции.

1. `strpbrk`
2. `strspn`
3. `strcspn`
4. `strchr`
5. `strrchr`

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

Задача 2

Номер задания = Номер в журнале % Количество вариантов.

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` или `strtok` нельзя).

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

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. 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.