

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

КУРСОВАЯ РАБОТА
по дисциплине «Программирование»
Тема: Обработка текстовых данных

Студент гр. 0304

Мажуга Д.Р

Преподаватель

Чайка К.В

Санкт-Петербург

2020

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент: Мажуга Д.Р.

Группа: 0304

Тема работы: Обработка текстовых данных

Исходные данные:

Программе на вход подается текст (текст представляет собой предложения, разделенные точкой. Предложения - набор слов, разделенные пробелом или запятой, слова - набор латинских букв, и цифр. Длина текста и каждого предложения заранее не известна. Программа должна сохранить этот текст в динамический массив строк и оперировать далее только с ним.

Содержание пояснительной записки:

Разделы пояснительной записки: «Содержание», «Введение», «Ввод текста», «Обработка текста», «Вывод текста», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки: не менее 10 страниц.

Дата выдачи задания: 02.11.2020

Дата сдачи реферата:

Дата защиты реферата:

Студент

Мажуга.Д.Р.

Преподаватель

Чайка К.В.

АНОТАЦИЯ

Была написана программа для обработки текста, длина которого изначально неизвестна. Текст состоит из слов, состоящих из латинских букв и цифр. Слова между собой разделены пробелом или запятой. Предложения разделяются точкой.

Программа предлагает пользователю ввести текст, после чего выводит список возможных операций. Пользователь может выбрать одну из предложенных операций либо выйти из программы. После обработки текст выводится на экран.

СОДЕРЖАНИЕ

Введение	5
1. Считывание текста	7
2. Обработка текста	7
2.1. Удаление повторяющихся предложений	7
2.2. Удаление первых слов во всех предложениях	8
2.3. Подсчёт времени в предложениях (конструкции типа «23sec»)	8
2.4. Удаление предложений в котором все слова одной длины.	8
2.5. Сортировка предложений по уменьшению количества строчных букв в нём	9
2.5.1. Функция count_alph()	9
2.5.2. Функция comp()	9
2.5.3. Сортировка предложению	9
3. Вывод текста	9
4. Меню	10
5. Вывод	10
6. Список использованных источников	11

ВВЕДЕНИЕ

Цель работы: создание программы, которая будет обрабатывать текст, введённый пользователем, в зависимости выбранной пользователем операции, а затем результат будет выведен на экран.

Условия:

Программе на вход подается текст (текст представляет собой предложения, разделенные точкой. Предложения - набор слов, разделенные пробелом или запятой, слова - набор латинских букв и цифр. Длина текста и каждого предложения заранее не известна.

Программа должна сохранить этот текст в динамический массив строк и оперировать далее только с ним.

Программа должна найти и удалить все повторно встречающиеся предложения (сравнивать их следует посимвольно, но без учета регистра).

Далее, программа должна запрашивать у пользователя одно из следующих доступных действий (программа должна печатать для этого подсказку. Также следует предусмотреть возможность выхода из программы):

1. Найти все предложения в которых есть слова вида “<число>sec” (“23sec”) задающие количество секунд. Для

каждого такого предложения посчитать количество секунд и вывести количество полученных часов, минут и секунд в виде строки "hour:min:sec".

2. В каждом предложении удалить первое слово. Если в предложении не останется слов, то его надо удалить.

3. Удалить все предложения у которых длины всех слов одинаковые.

4. Отсортировать предложения по уменьшению количества строчных букв в нем.

Все сортировки должны осуществляться с использованием функции стандартной библиотеки. Использование собственных функций, при наличии аналога среди функций стандартной библиотеки, запрещается.

Все подзадачи, ввод/вывод должны быть реализованы в виде отдельной функции.

Задачи для выполнения данной работы:

1. Разделение данного нам условия на отдельные функции.
2. Разработка программного кода, соответствующего условиям
3. Тестирование программного кода на наличие ошибок, их выявление и устранение

1. СЧИТЫВАНИЕ ТЕКСТА

Считывание пользовательского текста производится в функции `input`. Так как текст неизвестной длины, будет использовано динамическое выделение памяти. Сначала при помощи функции стандартной библиотеки `malloc()` выделяется память под первый символ, который будет записан в переменную-буфер `sentence`, и под первую строку. Затем происходит считывание символов, которые вводит пользователь, цикле `while`. Считывание заканчивается, когда пользователь вводит `'\n'` (символ переноса строки). Также внутри функции предусмотрены условие для увеличения памяти для предложения и действие при неудачном выделении памяти. В конце ввода происходит освобождение памяти, занятой `sentence`, при помощи функции `free()` стандартной библиотеки.

2. ОБРАБОТКА ТЕКСТА

2.1 Удаление повторяющихся предложений

Функция `delete_duplicate` находит во введенном тексте повторяющиеся предложения и удаляет их в независимости от выбора пользователя сразу после считывания текста.

Далее предложения сравниваются посимвольно без учёта регистра при помощи функции `strcasemp()`. Если найдены одинаковые предложения, то они удаляются следующим образом: на место предложения справа при помощи функции `memmove()` сдвигаются все остальные предложения, а память уменьшается при помощи функции стандартной библиотеки `realloc()`.

2.2 Удаление первых слов во всех предложениях

Функция `del_first_word()` проходится по тексту по символно, как только он натывается на пробел, запятую или точку, он копирует символы при помощи функции `memmove()`, которые стоят после пробела, на место первого слова, тем самым удаляя его. Функция возвращает изменённый текст.

2.3 Подсчёт времени в предложениях(конструкции типа «23sec»)

Функция `calc_time()` проходит посимвольно проверяя является ли символ числом, если да, то далее следует проверка стоит ли через три символа, символ 'с' (или через четыре, т.к число может быть двухзначным). Если все условия выполнены, то затем мы сдвигаем предложени при помощи функции `memmove()`, а далее при помощи функции `atoi()` преобразуем первое «слово» в число, сохраняя число в переменную `num`.

Далее вызываем функцию `transform_time()`, которая в свою очередь преобразует секунды в часы, минуты, и секунды, и выводит результат.

2.4 Удалени предложений в котором все слова одной длины.

Функция `del_len_sent()` при помощи цикла обходит текст по предложениям, копируя предложения в переменную `sentence` (при помощи функции `strcpy()`), для с помощью функции `strtok()` разделяем предложение на слова и сохраняем в массив строк `word_arr`.

Далее с помощью цикла, мы проверяем каждое слово на равенсто(по количеству символов) если они равны, то переменная `cnt` увеличивается на 1,

которая в последствии проверяется с количеством слов в массиве, если она равна количеству слов, то предложение удаляется. Функция возвращает изменённый текст.

2.5 Сортировка предложений по уменьшению количества строчных букв в нём

2.5.1 Функция count_alph()

Принимает на вход предложения, и обходит посимвольно, проверяя является ли символ в нижнем регистре или нет, если да, то увеличивает переменную count на 1, возвращает count.

2.5.2 Функция comp()

Вызывает функцию count_alph() и присваивает значение аргументам, далее сравнивая значения возвращает 1, -1, 0.

2.5.3 Сортировка предложению

При помощи функции qsort() происходит сортировка предложений текста. Функция возвращает изменённый текст.

3. Вывод текста

Функция output() выводит на экран пользователя обработанный программой текст при помощи цикла for.

4. Меню

В функции main() пользователю предлагается ввести текст при помощи функции input(). Если функции не удаётся выделить память и она возвращает NULL, то происходит завершение программы со значением 1.

Следом вызывается функция delete_duplicate(), удаляющая повторяющиеся предложения.

Далее программа выводит пользователю возможные действия с текстом с помощью значения n. После ввода выполняется цикл while. Внутри цикла

выполняется оператор switch, в зависимости от вводимого значения выполняется определенная функция: 0 – выход из функции; 1 – Удалить первое слово в каждом предложении; 2 – Удалить предложения в котором длины всех слов равны; 3 – Отсортировать предложения по количеству строчных букв в нём; 4 – Подсчёт времени в тексте; 5 - Распечатать текст. Если введена недоступная цифра, выполняется оператор default и будет выведено сообщение о недоступности операции. При любом выборе пользователь может выбрать другую операции и произвести повторную обработку текста. После выхода из программы, происходит освобождение памяти, и программа завершает свою работу.

Вывод

Была разработана программа, соответствующая поставленной цели. Данная программа считывает текст, введенный пользователем, может выполнить определённые операции над этим текстом в зависимости от выбора пользователя и вернуть измененный текст пользователю.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «Язык программирования Си» (англ. The C Programming Language) / Ритчи Деннис М., Керниган Брайан У. изд. Вильямс, 2017.
2. Онлайн справочник C/C++ // cplusplus.com URL:
<http://www.cplusplus.com>