

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

КУРСОВАЯ РАБОТА
по дисциплине «Программирование»
Тема: Структуры и динамическая память

Студент гр. 0303

Калмак Д.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2020

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Калмак Д.А.

Группа 0303

Тема работы: Структуры и динамическая память

Исходные данные:

Подается текст, необходимо сохранить его в динамический массив предложений, используя структуры. Осуществить функции на выбор пользователя.

Содержание пояснительной записки:

Аннотация, введение, структуры, ход работы, тестирование, инструкция, заключение, список использованных источников.

Предполагаемый объем пояснительной записки:

Не менее 00 страниц.

Дата выдачи задания: 02.11.2020

Дата сдачи реферата: 17.12.2020

Дата защиты реферата: 18.12.2000

Студент(ка)

Калмак Д.А.

Преподаватель

Чайка К.В.

АННОТАЦИЯ

Необходимо было разработать программу, которая на вход принимает текст, который необходимо обработать. Обработка осуществляется в зависимости от команд пользователя. Были реализованы структуры Sentence и Text для хранения текста и предложений. В результате программа может осуществлять:

первичную обработку – удаление повторных предложений.

Далее в зависимости от выбора пользователя происходят четыре обработки: сколько секунд в минутах встречается в тексте, сортировка предложений по возрастанию суммы кодов символов первых слов предложений, замена символов #-<решетка> @-(at) %-<persent>, удаление предложений с последним словом с тремя согласными подряд. А также реализован выход из программы.

СОДЕРЖАНИЕ

	Введение	5
1.	Структуры	7
1.1.	<i>Структура Sentence</i>	7
1.2.	<i>Структура Text</i>	7
2.	Ход работы	8
2.1.	<i>Реализация программы</i>	8
2.2.	<i>Сборка программы</i>	9
3.	Тестирование	10
4.	Инструкция	12
4.1	<i>Makefile</i>	12
4.2	<i>Работа программы</i>	12
	Заключение	13
	Список использованных источников	14

ВВЕДЕНИЕ

Цель работы: необходимо написать программу, используя структуры, принимающую на вход текст, который хранится в динамической памяти, и обрабатывающая его.

Задача:

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

Для хранения предложения и для хранения текста требуется реализовать структуры Sentence и Text

Программа должна сохранить (считать) текст в виде динамического массива предложений и оперировать далее только с ним. Функции обработки также должны принимать на вход либо текст (Text), либо предложение (Sentence).

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):

Посчитать и вывести в минутах количество секунд встречающихся в тексте. Количество секунд задается в виде подстроки “ <число> sec “.

Отсортировать предложения по увеличению суммы кодов символов первого слова в предложении..

Заменить все символы ‘%’, ‘#’, ‘@’ на “<percent>”, “<решетка>”, “(at)” соответственно.

Удалить все предложения которые заканчиваются на слово с тремя согласными подряд.

Все сортировки и операции со строками должны осуществляться с использованием функций стандартной библиотеки. Использование собственных функций, при наличии аналога среди функций стандартной библиотеки, запрещается.

Каждую подзадачу следует вынести в отдельную функцию, функции сгруппировать в несколько файлов (например, функции обработки текста в один, функции ввода/вывода в другой). Также, должен быть написан Makefile.

1. СТРУКТУРЫ

1.1. Структура Sentence

Поля:

int lensent – длина предложения.

wchar_t* sent – предложение.

int wordsymsum – сумма кодов символов первого слова.
предложения.

1.2. Структура Text

int amountsent – количество предложений.

struct Sentence* textsent – текст, состоящий из предложений.

2. ХОД РАБОТЫ

2.1. Реализация программы

Для функций реализован заголовочный файл со стандартными библиотеками функций языка Си.

Введенный текст делится на предложения, убираются при этом разделители между ними. Повторные предложения не записываются сразу, т.к. реализована посимвольная проверка без учета регистра.

Реализованы четыре обработки и выход из программы:

1. Подсчитывание секунд, которые встречаются в тексте, переводя в минуты:

Функция `countmin()`. Осуществляется поиск цифры, пробела, и подстроки “sec”. Циклом, идущим в обратную сторону, считываем цифры, умножая их на единицу, десятки и т.д. Далее осуществляется перевод в минуты и выводятся на экран.

2. Сортировка предложений по возрастанию суммы кодов символов первых слов предложений:

Функция `sortbyfirstword()` и функция, необходимая для функции `qsort()`, `compare()`. Сначала для каждого предложения высчитывается сумма символов первого слова, используется поле структуры `Sentence` – `wordsum`, так для каждого предложения хранится сумма кодов символов. Благодаря такому способу, используя функцию `qsort()`, можно сразу отсортировать предложения.

3. Замена символов #-<решетка> @-(at) %-<persent>:

Функции `replacetext()` и `replacesym()`. Вызывается функция `replacetext()`, которая для каждого предложения вызывает функцию `replacesym()`. В функции `replacesym()` определяется объект структуры `Sentence` `newsent`. С помощью цикла `for` программа проходит по всем элементам переданного предложения, и заполняет динамический массив `y` `newsent` – если символ не # @ %, то в массив записывается этот символ, иначе

записывается <решетка>, (at), <present> в зависимости от символа.

Новое предложение заменяет старое.

4. Удаление предложений с последним словом с тремя согласными подряд.

Функция `deletesentbylastword()`. Определяется объект структуры `Text newtext`. Создан массив `check` на столько элементов, сколько количество предложений. Далее циклом `for` проверяются каждые три символа последнего слова (цикл идет в обратную сторону) на вхождение в строку со всеми согласными буквами. Если предложение, в котором последнее слово с тремя идущими подряд согласными, то в массив записывается 1, а иначе 0. Идет проверка, если элемент массива `check` 0, то в `newtext` записывается это предложение. (сделано так, что индекс предложения и индекс `check` соответствуют друг другу).

5. Выход из программы, освобождается динамическая память под хранение текста из предложений.

2.2. Сборка программы

Функции ввода в отдельном файле и имеют заголовочный файл. Функции обработки в отдельном файле и имеют заголовочный файл. Создан общий заголовочный файл, в котором определены структуры и подключаются стандартные библиотеки (все единоразово). Создан `Makefile`: сначала компилируются объектные файлы, затем создается исполняемый файл.

3. ТЕСТИРОВАНИЕ

Удалено повторяющееся предложение (I am seven.). (30+50+11)/60 верно (посчитаны минуты).

```
My name is K@te. Your surname 30 sec is Pavlova. I am seven. I live in Minsk, in Pushkin 50 sec street. I go to school number 214. I am seven. I am in the 11 sec fi#rst formm. I go to school number 214.
Вывод разных предложений:
My name is K@te.
Your surname 30 sec is Pavlova.
I am seven.
I live in Minsk, in Pushkin 50 sec street.
I go to school number 214.
I am in the 11 sec fi#rst formm.
I go to school number 214.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
1
Результат кодового значения - 1:
Количество минут: 1,516667
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
5
Выход из программы.
```

Удалено повторяющееся предложение (I am seven.). (Сортировка 73,73,73,73,73,198,331)

```
My name is K@te. Your surname 30 sec is Pavlova. I am seven. I live in Minsk, in Pushkin 50 sec street. I go to school number 214. I am seven. I am in the 11 sec fi#rst formm. I go to school number 214.
Вывод разных предложений:
My name is K@te.
Your surname 30 sec is Pavlova.
I am seven.
I live in Minsk, in Pushkin 50 sec street.
I go to school number 214.
I am in the 11 sec fi#rst formm.
I go to school number 214.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
2
Результат кодового значения - 2:
I am seven.
I live in Minsk, in Pushkin 50 sec street.
I go to school number 214.
I am in the 11 sec fi#rst formm.
I go to school number 214.
My name is K@te.
Your surname 30 sec is Pavlova.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
5
Выход из программы.
```

Удалено повторяющееся предложение (I am seven.). Заменены символы(#,%,@).

```
My name is K@te. Your surname 30 sec is Pavlova. I am seven. I live in Minsk, in Pushkin 50 sec street. I go to school number 214. I am seven. I am in the 11 sec fi#rst formm. I go to school number 214.
Вывод разных предложений:
My name is K@te.
Your surname 30 sec is Pavlova.
I am seven.
I live in Minsk, in Pushkin 50 sec street.
I go to school number 214.
I am in the 11 sec fi#rst formm.
I go to school number 214.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
3
Результат кодового значения - 3:
My na<persent>me is K(at)te.
Your surname 30 sec is Pavlova.
I am seven.
I live in Minsk, in Pushkin 50 sec street.
I go to sch<решетка>ool number 214.
I am in the 11 sec fi<решетка>rst formm.
I go to school number 214.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
5
Выход из программы.
```

Удалено повторяющееся предложение (I am seven.). Удалены предложения с тремя подряд идущими согласными в последнем слове.

```
My name is K@te. Your surname 30 sec is Pavlova. I am seven. I live in Minsk, in Pushkin 50 sec street. I go to school number 214. I am seven. I am in the 11 sec fi#rst formm. I go to school number 214.
Вывод разных предложений:
My name is K@te.
Your surname 30 sec is Pavlova.
I am seven.
I live in Minsk, in Pushkin 50 sec street.
I go to school number 214.
I am in the 11 sec fi#rst formm.
I go to school number 214.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
4
Результат кодового значения - 4:
My name is K@te.
Your surname 30 sec is Pavlova.
I am seven.
I go to school number 214.
I go to school number 214.
Введите кодовое значение от 1 до 4 (для выхода нажмите 5):
5
Выход из программы.
```


4. ИНСТРУКЦИЯ

4.1 Makefile:

Находясь в папке с файлами, прописать команду make.

4.2 Работа программы:

Прописать команду ./sw

На вход поступают предложения. Признак конца ввода – два пробела.
Удаляются одинаковые предложения. Далее на выбор пять действий программы:

1. Посчитать в минутах количество секунд, встречающихся в тексте.
2. Отсортировать предложения по возрастанию суммы кодов символов первого слова предложений.
3. Заменить в тексте:
% - <present>
- <решетка>
@ - (at)
4. Удалить предложения, оканчивающиеся на слово с тремя согласными подряд.
5. Выход из программы.

ЗАКЛЮЧЕНИЕ

Таким образом, текст был сохранен в динамический массив предложений, используя структуры. Реализованы функции для обработки текста на выбор пользователя.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Б.В. Керниган, Д.М. Ричи. «ЯЗЫК С»*
2. *Структуры в си.*
<https://learnc.info/c/structures.html>