

**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

\_\_\_\_\_

Докучаев Р.А.

Преподаватель

\_\_\_\_\_

Чайка К.В.

Санкт-Петербург

2020

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Докучаев Р.А.

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:

«Содержание», «Введение», «Считывание исходного текста», «Обработка текста», «Вывод обработанного текста», «Взаимодействие с пользователем», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее страниц.

Дата выдачи задания: 2.11.2020

Дата сдачи реферата: 23.12.2020

Дата защиты реферата: 24.12.2020

Студент

\_\_\_\_\_

Докучаев Р.А.

Преподаватель

\_\_\_\_\_

Чайка К.В.

## **АННОТАЦИЯ**

Программе была поставлена задача обработать поданный пользователем исходный текст и вывести его на экран. Текст – набор предложений. Предложение – набор латинских символов, разделенных пробелом или запятой. Предложение заканчивается точкой. Длина текста или каждого предложения заранее неизвестна.

Программа предлагает пользователю ввести текст, затем сохраняет его в динамический массив строк и оперирует с ним. После этого программа предлагает пользователю выбрать одно из предложенных действий по обработке текста. После завершения действия программа выводит обработанный текст пользователю.

## СОДЕРЖАНИЕ

	Введение	5
1.	Считывание текста	7
2.	Обработка текста	7
2.1.	Удаление повторяющихся предложений	7
2.2.	Подсчёт всех цифр, встречающихся в тексте	7
2.3.	Вывод текста в обратном порядке без учёта разделителей	8
2.4.	Удаление предложений, содержащих слово “physics”	8
2.5.	Сортировка	8
2.5.1.	Функция для сравнения строк	8
2.5.2.	Функция сортировки	9
3.	Вывод текста	9
4.	Меню (Взаимодействие с пользователем)	9
4.1.	Функция main	9
	Заключение	10
	Список использованных источников	11

## ВВЕДЕНИЕ

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

**Исходные 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. Вывести список всех цифр, встречаемых во всем тексте и их частоту.
2. Преобразовать предложение так, чтобы символы кроме разделительных шли в обратном порядке. Например, для строки “abc defg.” результатом будет “gfe dcba.”.
3. Удалить все предложения в которых встречается слово “physics”.
4. Отсортировать предложения по уменьшению количества слов длина которых равняется 3.

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.

Для достижения поставленной цели потребуется реализовать следующие задачи:

1. Разделение исходного 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 *strcasestr()*. 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. Подсчёт всех цифр, встречающихся в тексте

Функция *is\_digit* проходится по тексту посимвольно при помощи цикла *for* и проверяет, является ли символ цифрой. Результат сохраняется в динамический массив целых чисел *arr[]*.

Проверка происходит последовательно для каждой цифры при помощи функции *isdigit()*. В случае, если какая-то цифра присутствует в тексте, элемент массива, соответствующий её порядковому номеру, увеличивается.

### 2.3. Вывод текста в обратном порядке без учёта разделителей

Функция *reverse\_order* проходится по каждому предложению посимвольно.

При помощи двух переменных *begin* (указывает на первый символ предложения) и *end* (указывает на последний символ предложения до точки) происходит обмен символов (если это не разделители (запятая или пробел)) до тех пор, пока они не примут одинаковые значения.

### 2.4. Удаление предложений, включающих слово “physics”

Функция *delete\_physics* удаляет все предложения, в которых есть слово “physics”.

При помощи функции *strstr()* в каждом отдельном предложении текста будет найдено слово “physics”. В случае если предложение содержит это слово, то оно будет удалено аналогичным функции *delete\_duplicate* образом: на место предложения справа при помощи функции *memmove()* сдвигаются все остальные предложения, а память уменьшается при помощи функции стандартной библиотеки *realloc()*.

### 2.5. Сортировка

#### 2.5.1. Функция для сравнения строк

Функция *function\_for\_sort\_sentence* сравнивает два предложения и в зависимости от значения двух переменных-счётчиков *count* возвращает основания для сравнения двух предложений: -1, 0, 1.

При помощи указателей подсчитывается количество слов из 3 букв в каждом из предложений, которые будут найдены при помощи флага и разность между началом слова и текущим символом. В случае если найдено слово из 3 букв, переменная-счётчик увеличивается на 1. В конце происходит сравнение двух счётчиков. В зависимости от результата сравнение, происходит сортировка.



### 2.5.2. Функция сортировки

Функция *sorting\_function* сортирует все предложения текста при функции для сравнения строк по количеству слов из 3 букв.

При помощи функции быстрой сортировки *qsort* происходит сортировка предложений текста. Функция возвращает измененный текст.

## 3. ВЫВОД ТЕКСТА

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

## 4. МЕНЮ (ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ)

### 4.1. Функция *main()*

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 значение *key*. После ввода выполняется цикл *while*. Внутри цикла выполняется оператор *switch*, в зависимости от вводимого значения выполняется определенная функция: 0 – выход из функции; 1 – вывести массив всех цифр текста; 2 – преобразовать предложение так, чтобы символы кроме разделительных шли в обратном порядке; 3 – удалить все предложения со словом *physics*; 4 – отсортировать предложения текста по количеству слов из 3 букв. 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.

## **ЗАКЛЮЧЕНИЕ**

Была разработана программа, соответствующая поставленной цели. Данная программа считывает текст, введенный пользователем, может выполнить определённые операции над этим текстом в зависимости от выбора пользователя (вывести список всех цифр в тексте, вывести текст в обратном порядке, удалить все предложения с определённым словом или отсортировать текст по указанному признаку) 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>