

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Программирование»
Тема: Указатели и массивы.

Студент гр. 1382

Коренев Д.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Освоение работы с указателями и динамической памятью.

Задание (Вариант 4).

Напишите программу, которая форматирует некоторый текст и выводит результат на консоль.

На вход программе подается текст, который заканчивается предложением "Dragon flew away!".

Предложение (кроме последнего) может заканчиваться на:

. (точка)

; (точка с запятой)

? (вопросительный знак)

Программа должна изменить и вывести текст следующим образом:

Каждое предложение должно начинаться с новой строки.

Табуляция (`\t`, `' '`) в начале предложения должна быть удалена.

Все предложения, в которых есть число 555, должны быть удалены.

Текст должен заканчиваться фразой "Количество предложений до n и количество предложений после m ", где n - количество предложений в изначальном тексте (без учета терминального предложения "Dragon flew away!") и m - количество предложений в отформатированном тексте (без учета предложения про количество из данного пункта).

* Порядок предложений не должен меняться

* Статически выделять память под текст нельзя

* Пробел между предложениями является разделителем, а не частью какого-то предложения.

Выполнение работы.

Подключение необходимых библиотек, а так же, с помощью директивы `define`, определение значения для `SIZE` и `STEP`.

В функции `main` определяется «стоп слово». Выделяется динамическая память типа `char*` для хранения указателей, прочие переменные. В цикле `while` считывается строка при помощи функции

read_text(), а также удаляется табуляция в начале и само предложение, если в нем есть число 555, функцией delete_tabul_and_fives_words(). Идет проверка условия: если полученная строка является стоп словом, цикл завершается при помощи break. Если после добавления указателя на предложения в массив указателей предложений (text_arr) в нем не осталось места, оно выделяется при помощи функции realloc(). Далее, в цикле for, выводятся все предложения на экран через разделитель “\n” и последнее предложение, в котором говорится о количестве предложений до удаления некоторых из них и после удаления.

Функция read_text() динамически выделяет память типа char. В цикле while считывает символ, который поступил на вход и добавляет его в массив символов (text). Если после добавления не осталось свободного места, память выделяется дополнительно. Если этот символ был символом конца предложения, то цикл завершится. После последнего символа в массив добавляется «\n» - конец предложения, и возвращается указатель на этот массив.

Функция delete_tabul_and_fives_words() принимает в качестве аргумента указатель на предложение. В цикле while если первый символ является символом табуляции, то все предложение смещается «влево» на 1 (элемент с индексом 2 будет стоять под индексом 1 и тд). При помощи функции strstr можно определить, есть ли в предложении подстрока "555" и если есть, определяется, стоит ли пробел или символ конца предложения после него, если есть, то либо «555» в начале предложения или перед ним есть пробел. Если одно из условий выполнилось, функция возвращает указатель на массив, в котором есть только символ конца предложения, иначе – указатель на массив без табуляции в начале.

Тестирование.

Результаты тестирования представлены в таблице 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1	Test sentence555? Yes 555; ok. Dragon flew away!	Test sentence555? ok. Dragon flew away! Количество предложений до 3 и количество предложений после 2	Программа работает верно.
2	One more test 555; it must work? Some words without meaning. Dragon flew away!	it must work? Some words without meaning. Dragon flew away! Количество предложений до 3 и количество предложений после 2	Программа работает верно.
3	Last test without fives ? rpfirfi ifninsa; dfpd wfs sdfw. Wer? Dragong flew away!	Last test without fives ? rpfirfi ifninsa; dfpd wfs sdfw. Wer? Dragong flew away! Количество предложений до 4 и количество предложений после 4	Программа работает верно.

Выводы.

Были изучены указатели и динамические массивы на языке Си.
Разработана программа с использованием указателей и динамической

памяти. Программа была разделена на функции для легкости чтения кода и возможности менять ее локально.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название программы: PR_Korenev_Danil_lb3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 50
#define STEP 10

char* read_text(){
    int len = 0, size = SIZE, c;
    char* text = malloc(size*sizeof (char));

    while(1){
        c = getchar();

        text[len] = c;
        len++;
        if (len == size){
            size += STEP;
            text = (char*) realloc(text, size);
        }
        if (c == '.' || c == ';' || c == '?' || c == '!'){
            break;
        }
    }
    text[len] = '\0';
    return text;
}

char* delete_tabul_and_fives_words(char* text){
    int len = strlen(text);
    while(text[0] == ' ' || text[0] == '\t' || text[0] == '\n'){
        for(int count = 0; count < len; count++){
            text[count] = text[count+1];
        }
    }

    // если 555 в строке text[0] = '\0'
    char fives[] = "555";
    char* num_fivs = strstr(text, fives);
    if (num_fivs){
        // после 555 пробел или конец предложения
        if (num_fivs[3] == ' ' || num_fivs[3] == '.' ||
num_fivs[3] == ';' || num_fivs[3] == '?' || num_fivs[3] == '!'){
            //если 555 в начале текста
            if (strlen(text) == strlen(num_fivs)){
                text[0] = '\0';
            }else{
                //не в начале но будет пробел перед 555
                if (text[strlen(text) - strlen(num_fivs) - 1] ==
' '){
                    text[0] = '\0';
                }
            }
        }
    }
}
```

```

        }
    }
}

return text;
}

int main(){
    char* stop_word = "Dragon flew away!";
    char** text_arr = (char**)malloc(SIZE*sizeof(char*));
    int text_len = 0; //количество предложений
    int size_text_arr = SIZE;
    char* str; //слово
    int finish_len = 0;

    while (1){
        str = read_text();
        str = delete_tabul_and_fives_words(str);
        if (!strcmp(text_arr[text_len++] = str, stop_word)){
            break;
        }
        if (text_len == size_text_arr){
            size_text_arr += STEP;
            text_arr = (char**) realloc(text_arr,
size_text_arr*sizeof(char*));
        }
    }

    for (int i = 0; i < text_len; i++){
        if (strcmp(text_arr[i], "\0")){
            fputs(text_arr[i], stdout);
            printf("\n");
            finish_len++;
        }
    }
    printf("Количество предложений до %d и количество
предложений после %d\n", text_len-1, finish_len-1);
    free(text_arr);
    return 0;
}
}

```