

**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:

Использование указателей

Студент гр. 2384

Кузьминых Е.М.

Преподаватель

Гаврилов А.В.

Санкт-Петербург

2022

Цель работы.

Написать программу на языке программирования C, которая будет принимать текст неизвестной длины, редактировать его и выводить измененный текст, научиться работать с указателями в C, а также обрабатывать текст.

Задачи.

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 в начале предложения должна быть удалена.

Все предложения, в которых больше одной заглавной буквы, 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.

В начале программы мы подключим библиотеки `<stdio.h>` `<stdlib.h>` `<string.h>` `<ctype.h>`, а также объявим макроопределение `TEXT_SIZE` для выделения памяти.

Программу разобьем на несколько функций. Так, функция `read_sentence()` будет считывать предложение. В начале мы выделяем память для нашего предложения и начинаем посимвольно считывать его, а также инициализируем переменную `size` – размер буфера для предложения и `sentence_len` – длину предложения. Первый символ мы проверяем, не является ли он пробелом, символом `'\t'` или `'\n'` и в случае совпадения игнорируем его, чтобы удалить отступы в тексте, иначе добавляем его к строке. После в цикле `do while` мы посимвольно считываем предложение, проверяем, хватает ли нам памяти (сравнение буфера и длины предложения) и в случае нехватки памяти перевыделяем с помощью функции `realloc()`. Текст мы считываем до момента, пока не встретим символ препинания (точку, восклицательный или вопросительный знак). Затем последний индекс в предложении мы делаем равным `'\0'` для «завершения» строки и возвращаем считанное предложение.

В функции `count_upper_symbols()` мы принимаем предложение и считаем количество букв в верхнем регистре. В случае, если количество таких символов меньше или равно 1, то функция вернет 1, иначе – 0.

В `main()` мы по аналогии с функцией `read_sentence()` инициализируем `text_size` – буфер для нашего текста, `count_sentence` – количество предложений в тексте, `char** text` – текст, куда мы будем сохранять считанные предложения `count_deleaded_sentences` – количество удаленных предложений, `sentence` – текущее предложение для считывания и `end_of_text` – «предложение -флаг», после которого считывание текста останавливается. Затем, с помощью цикла `do while` мы считываем предложение с помощью функции `read_sentence()`, проверяем его на количество символов в верхнем регистре, если

count_upper_symbols() возвращает 1, то мы записываем это предложение в *text*, иначе увеличиваем *count_dealed_sentences* на 1. В случае равенства буфера для памяти и количества предложений (*count_sentence* и *text_size*) мы увеличиваем буфер и перевыделяем память. Считываем текст до того момента, пока считанное предложение не будет совпадать с *end_of_text*. После окончания ввода мы с помощью цикла *for* «проходим» по каждому предложению в *text*, выводим его и освобождаем его память. После освобождения всей памяти мы выводим количество предложений до обработки (равное *count_sentence+count_dealed_sentences-1*) и количество предложений после обработки (*count_sentence -1*).

Тестирование.

№	Входные данные	Выходные данные	Комментарий
1	LLLorem impuls. Hello everyone. Dragon flew away!	Hello everyone. Dragon flew away! Количество предложений до 2 и количество предложений после 1	Ответ верный
2	First sentence. SECOND sentence. Third sentence. Dragon flew away!	First sentence. Third sentence. Dragon flew away! Количество предложений до 3 и количество предложений после 2	Ответ верный

Вывод.

В ходе выполнения лабораторной работы была изучена работа с указателями, методы работы со строками. Была написана программа, принимающая текст неизвестной длины, которая считывает текст до предложения “Dragon flew away!”, удаляет отступы в начале предложения, выводит измененный текст, а также считает количество предложений в тексте до обработки и после.

Приложение 1. Исходный код программы.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <malloc.h>
#include <stdlib.h>
#define TEXT_SIZE 100
//чтение предложения
char* read_sentence(){
    int size = TEXT_SIZE;
    char *sentence = malloc(size*sizeof(char));
    int sentence_len = 0;
    int symbol = getchar();
    if (symbol == '\n' || symbol=='\t' || symbol==' ');
    else{
        sentence[sentence_len++]=symbol;
    }
    do{
        symbol = getchar();
        sentence[sentence_len++]=symbol;
        if(sentence_len==size){
            size+=TEXT_SIZE;
            sentence= realloc(sentence,size);
        }
    }while(symbol!='!'&& symbol!='.' &&symbol!=';'&& symbol!='?');
    sentence[sentence_len]='\0';
    return sentence;
}
//количество заглавных символов в предложении, больше 1 - false, меньше
- true
int count_upper_symbols(char *sentence){
    int count = 0;
    for(int i=0; i< strlen(sentence); i++){
        if(isupper(sentence[i])){
            count++;
        }
    }
    if (count>1){
```

```

        return 0;
    }else{
        return 1;
    }
}

int main(){
    int text_size= TEXT_SIZE;
    char** text = malloc(text_size*sizeof(char*));
    char* end_of_text = "Dragon flew away!";
    char* sentence;
    int count_sentence = 0;
    int count_deleaded_sentences = 0;

    do {
        sentence= read_sentence();
        if(count_upper_cymbols(sentence)){
            text[count_sentence++]=sentence;

        }else{
            count_deleaded_sentences+=1;
        }
        if(count_sentence==text_size){
            text_size+=TEXT_SIZE;
            text =realloc(text,text_size*sizeof (char*));
        }
    } while (strcmp(sentence,end_of_text)!=0);

    for (int i = 0; i < count_sentence; i++) {
        puts(text[i]);
        free(text[i]);
    }
    free(text);
    printf("Количество предложений до %d и количество предложений после %d",count_sentence+count_deleaded_sentences-1,count_sentence-1);
    return 0;
}

```