

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. 0304

Докучаев Р.А.

Преподаватель

Чайка К.В.

Санкт-Петербург

2020

Цель работы.

Изучить работу указателей, динамической памяти и строк в языке Си и написать программу, которая считывает и обрабатывает текст.

Задание.

Вариант №2.

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: точку, точку с запятой или 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, которые заканчиваются на '?' 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

Основные теоретические положения.

Использованы стандартные заголовочные файлы *stdio.h*, *stdlib.h*, заголовочный файл для работы со строками *string.h*, функции *malloc* и *realloc* для выделения памяти под динамический массив, функция *free* для освобождения памяти после работы с массивом. Также для работы с указателями использован оператор ***.

Выполнение работы.

1. Подключение заголовочных файлов *stdio.h*, *stdlib.h* и *string.h*.
2. Создание символической константы *size_sent*, которая обозначает размер буфера.
3. Инициализация переменных и указателей для работы с первоначальным текстом: *char* sentence* – массив, хранящий предложения первоначального текста, *char symbol* – массив, хранящий символы введенных предложений, *int n*, *m* – счётчики предложений до редактирования и после, *int a* – переменная-счётчик для номера символа в предложениях, *int i* – переменная, равная *size_sent* для дальнейшего возможного увеличения выделенной динамической памяти.
4. Выделение динамической памяти для буфера предложения при помощи функции *malloc* из стандартной библиотеки.
5. Написание широкого цикла *while*, который включает в себя следующие действия с исходным текстом:
 - 1) считывание первого символа;
 - 2) пропуск табуляций и пробелов между предложениями в исходном тексте при помощи вложенного цикла *while*;
 - 3) выполнение посимвольного считывания при помощи вложенного цикла *while*, также в этот цикл включено условие переполнения, в случае которого при помощи функции стандартной библиотеки *realloc* происходит расширение динамической памяти;
 - 4) записываем в массив предложений последний символ
 - 5) условие выхода из цикла, реализованное с помощью вложенного *if*;
 - 6) подсчёт предложений после редактирования и их вывод с помощью вложенного *if*;
 - 7) подсчёт предложений до редактирования при помощи *инкремента*;
 - 8) перевод переменной-счётчика *a* в первоначальное состояние.
6. Вывод на экран количества предложений в тексте до редактирования и после при помощи функции стандартной библиотеки *printf*.
7. Освобождение памяти, которую занимал буфер при помощи функции стандартной библиотеки *free*.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Dragon flew away!	Dragon flew away! Количество предложений до 0 и количество предложений после 0	ОК
2.	Roma - author; Aizek - friend? Konstantin - master. Dragon flew away!	Roma - author; Konstantin - master. Dragon flew away! Количество предложений до 3 и количество предложений после 2	ОК
3.	Zoya - my love. I like her very much. She is beatiful and I want to live with her. Mum, what do you think about it? Dragon flew away!	Zoya - my love. I like her very much. She is beatiful and I want to live with her. Dragon flew away! Количество предложений до 4 и количество предложений после 3	ОК
4.	Пример, указанный в условии задачи в онлайн-курсе	Результат, полученный в ходе выполнения программы совпа- дает с результатом в онлайн- курсе	ОК

Выводы.

Мы изучили работу указателей, динамических массивов и строк в языке Си.

Мы написали программу, которая считывает исходный текст до терминальной фразы и преобразует его согласно условию задания.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название исходного файла: laba3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define size_sent 100

int main(){
char* sentence;
char symbol;
int n = 0, m = 0, a = 0, i = size_sent;
//Выделяем память под предложение
sentence = (char*) malloc(size_sent * sizeof(char));
while (1){
    //Считываем первый символ
    symbol = getchar();
    //Пропуск табуляций и пробелов
    while (symbol == '\t' || symbol == ' '){
        symbol = getchar();
    }

    while (symbol != '.' && symbol != ';' && symbol
!= '?' && symbol != '!' ){
        //Считываем посимвольно предложения из исходного тек-
ста
        sentence[a] = symbol;
        a++;
        if (a >= i - 2){
            //Текущий индекс вышел за размер буфера => увеличи-
ваем память
            i += size_sent;
            sentence = (char*) realloc(sentence, i
* sizeof(char));
        }
        symbol = getchar();
    }
    sentence[a] = symbol;
    sentence[a + 1] = '\0';
    //Записываем последний символ в предложений
    //Выходим из цикла, если
    if (symbol == '!'){
        printf("%s\n", sentence);
        break;
    }
}
```

```

        //Подсчёт предложений после редактирования и вы-
        вод предложений, которые заканчиваются не на вопроси-
        тельный знак
        if (symbol == '.' || symbol == ';'){
            m++;
            printf("%s\n", sentence);
        }
        //Подсчёт предложений до редактирования
        n++;
        a = 0;
    }
    printf("Количество предложений до %d и количество
    предложений после %d\n", n, m);
    //Вывод количества предложений
    free(sentence);
    return 0;
}

```