

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

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Программирование»**  
**Тема: «Использование указателей»**

Студент гр. 7381

\_\_\_\_\_

Дорох С.В.

Преподаватель

\_\_\_\_\_

Берленко Т.А.

Санкт-Петербург

2017

## Цель работы:

Написать 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, 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.

## Основные теоретические положения:

На вход программе поступает строка символов. Она считывается при помощи функции `getchar`. Далее под символы динамически выделяется память. И данные символы записываются в массив указателей. Потом происходит проверка, является ли символ точкой, точкой с запятой, вопросом или восклицательным знаком и переменной `flag` присваивается значение 1. Также сразу происходит проверка на наличие пробелов и табуляций в строке после символов: "?" ; "," ; "." ; "!" если таковые имеются ,то программа их игнорирует. Переменная `after` увеличивается на значение, которое возвращает функция `pr_str` (возвращает 0 только в том случае, если символ строки равен "?"), переменная `before` увеличивается на единицу при встрече с точкой, точкой с запятой. вопросом и восклицательным знаком. Функция `pr_str` проверяет не является ли символ

строки вопросом, если это так, то она присваивает переменной true единицу, в противном случае true равен 0. Если true равен единице то, функция pr\_str выводит строку до данного символа(включая и его самого) при помощи функции printf с новой строки. Текст заканчивается фразой «Количество предложений до n и количество предложений после m», где n- количество предложений в изначальном тексте (без учёта терминального “Dragon flew away!”) и m – количество предложений в отформатированном тексте. Также была освобождена память при помощи функции free.

## **Вывод:**

В данной лабораторной работе была изучена работа с указателями, динамическое выделение памяти, работа с данной памятью. Были изучены алгоритмы, позволяющие отформатировать текст согласно задаче.

## **Исходный код:**

### **Функция main.c:**

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

```
int pr_str(char *str, int current){
int true=1;
if (str[current-1]=='?'){
    true = 0;

}
```

```
if (true){  
    for (int i=0;i<current;i++){  
        printf("%c", str[i]);  
    }  
    printf("\n");  
    return 1;  
}  
return 0;  
}
```

```
int main(){  
    int i=0;  
    int length=0;  
    char n=0;  
    int current=0;  
    int before=0;  
    int after=0;  
    int flag=1;  
    char *str = malloc(2000* sizeof (char));
```

```
    while ((n=getchar()) != '\n' ){  
        if (flag){  
            if (n==' ' || n=='\t'){  
                continue;  
            }  
            flag=0;
```

```
}
```

```
length=current;
```

```
*(str+length)=n;
```

```
current=current+1;
```

```
if (n=='.' || n==';' || n=='?' || n=='!'){
```

```
after=after + pr_str(str, current);
```

```
current=0;
```

```
flag=1;
```

```
before++;
```

```
}
```

```
}
```

```
printf ("Количество предложений до %d и количество предложений  
после %d", (before-1), (after-1));
```

```
free (str);
```

```
return 0;
```

```
}
```

### **Makefile:**

```
All: main.o
```

```
    Gcc main.o
```

```
Main.o: main.c
```

```
    Gcc -c main.c
```