

**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.**

Студентка гр. 7381

Аженилок В. А.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2017

## Цель работы

Познакомиться с указателями, строками, динамической памятью, а также с функциями для работы с ними.

## Задание:

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, 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).

## Дополнительные условия:

- \* 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

1. <string.h> содержит прототип функции strlen(возвращает длину строки);
2. <stdio.h> – содержит прототипы функций ввода и вывода printf, puts, getchar

3. <stdlib.h> содержит прототипы функций calloc(выделяет блок памяти для массива размером num элементов, каждый из которых занимает size байт, и инициализирует все свои биты в нулями.), realloc(изменяет величину выделенной памяти), malloc(выделяет блок памяти, размером sizeof(int) байт, и возвращает указатель на начало блока.);
4. <ctype.h> содержит прототип функции isupper( проверяет символ, передаваемый через параметр, является ли он прописным символом);

### Вывод:

В процессе выполнения лабораторной работы были изучены понятие указателя, синтаксис его объявления, а также использование. Изучены понятия динамической памяти, функции для работы с ней в С (выделение через malloc, calloc, realloc и освобождение через free). Изучено представление строк в С, а так же методы работы с ними.

### Исходный код проекта:

#### 1.Makefile

```
all: main.o
    gcc main.o

main.o: main.c
    gcc -c main.c

clean:
    rm main.o
```

#### 2.main.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
    int i=0;
    int len=1;
    char* sentence=calloc(len, sizeof(char));
    int len1=1;
    char a,b;
    while((b=getchar())!=EOF)
    {
        ++len;
        sentence=realloc(sentence, len*sizeof(char));
```

```

        sentence[len-2]=b;
    }
    sentence[len-1]='\0';
    char* sentence2=calloc(len1, sizeof(char));
    int m=0,n=0,k=0;

    int Up=0;

    for (i=0;i<strlen(sentence);++i){
        a=sentence[i];
        sentence1[k]=a;
        if((sentence1[0]==' ')||(sentence1[0]=='\t')){
            ++i;
            a=sentence[i];
            sentence1[0]=a;
        }
        ++k;
        ++len1;
        sentence1=(char*)realloc(sentence1,len1*sizeof(char));
        if(isupper(a)){++Up;}
        if ((a == '.') || (a == ';') || (a == '?') || (a == '!')){
            ++m;
            ++n;
            if(Up<=1){
                sentence1[k]='\0';
                printf("%s\n",sentence1);}
            else{--m;}
        }
        Up=0;
        k=0;
        len1=1;
        sentence1=(char*)realloc(sentence1,len1*sizeof(char));
    }

    printf("Количество предложений до %d и количество предложений после
%d ",n-1,m-1);
    free(sentence);
    free(sentence1);
    return 0;
}

```