

**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

Кортев Ю.В.

Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

Берленко Т.А

Санкт-Петербург

2017

## Цель работы.

Освоение практического применения указателей и динамической памяти.

## Задача.

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 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

## Ход работы.

1. Динамически выделяю память под текст, размером 30 ячеек типа char.
2. Если массив заполняется, то функция realloc увеличивает размер массива еще на 30 ячеек.
3. Также при заполнении массива text идет подсчет предложений.

4. Динамический массив text\_out инициализируется и заполняется в каждой ветви цикла for, кол-во ветвей зависит от кол-ва предложений в тексте.
5. Text\_out заполняется из массива text, убирая табуляцию.
6. После переписывания предложения из text в text\_out, массив печатается на экран с символом переноса строки.
7. Если в предложении содержится 3 идущих подряд цифр 5, предложение не печатается, а цикл переходит к следующей ветви.

### **Вывод.**

Выполнив данную лабораторную работу, я освоил практическое применение указателей, научился работать со строками.

### **Исходный код.**

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

int main() {
    int i;
    int n=0;
    int m=0;
    char c;
    int del;
    int d=1;
    int f=0;
    char* text_out;
    char* text=(char*)malloc(30*sizeof(char));
    while(c=getchar()!='\n')
    {
        if(c=='.' || c==',' || c=='?')//кол-во предложений
        {
            n++;
        }
        if(i==(30*d))
        {
            d++;
            text=realloc(text, (30*d)*sizeof(char));
        }
    }
}
```

```
text[i]=c;
i++;
}
```

```
//фильтр
char *print_out;
for(i=0;i<n;i++)
{
    text_out=malloc((strlen(text)+1)*sizeof(char));
    while(*text!='.'|| *text!=';' || *text!='?')
    {

        print_out=text_out;
        if(*text=='\t')
        {
            text++;
        }
        if((f==1||f==2)&*text!='5')
        {
            f=0;
        }
        if(*text=='5')
        {
            f++;
        }
        if(f==3)
        {
            m++;
            del=1;
        }

        *text_out=*text;
        text++;
        text_out++;
    }
    *text_out=*text;
    if(!del)
    {
        printf("%s\n", print_out);
    }
    else
```

```
    {del=0;  
    }  
    free(text_out);  
    free(print_out);  
    text++;  
}  
printf("%s", text);
```

```
free(text);
```

```
    printf("Количество предложений до %d и количество  
предложений после %d", n, n-m);
```

```
return 0;  
}
```