

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

Ильясов А. В.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2017

Цель работы:

Научиться работать с указателями, строками, динамической памятью и функциями для работы с ними.

Задача:

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).

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 sizetem 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 указатели, работа с динамической памятью, функции для работы с ней (выделение с помощью функций malloc, calloc, realloc и освобождение с помощью функции free). Изучено представление строк в C, а так же методы работы с ними.

Исходный код:

main.c

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define N 5000

int print_str(char *str, int pos) {

    int true = 1;
    if (str[pos-1] == '?') {
```

```

        true = 0;
    }

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

    int i = 0;
    int t = 0;
    int len;
    char c;
    int pos = 0;
    int n = 0;
    int m = 0;
    int flag = 1;
    char *str = malloc(N*sizeof(char));

    while ((c = getchar()) != '\n') {

        if (flag) {
            if (c == ' ' || c == '\t') {
                continue;
            }
            flag = 0;
        }

        len = pos;
        str[len] = c;
        pos++;
        if (c == '.' || c == ';' || c == '?') {
            m = m + print_str(str, pos);
            pos = 0;
            flag = 1;
            n++;
        }
    }

    printf ("Количество предложений до %d и количество предложений после %d\n", n, m);
    free (str);
return 0;
}

```