**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. 6304 | Попов С.С. |  |
| Преподаватель | Кринкин К.В. |  |

Санкт-Петербург

2016

**Цель работы**

Написать программу, которая форматирует некоторый текст и выводит результат на консоль.

**Задание**

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:

* Все предложения, которые заканчиваются на '?' должны быть удалены.
* Каждое предложение должно начинаться с новой строки.
* Табуляция в начале предложения должна быть удалена.
* 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**

Используйте несколько функций для обработки текста.

Лабораторная работа №3 должна быть выложена на Github, аналогично 2й.

**Содержание**

**Вариант со статическим выделением памяти**

Text\_v2.c

#include "stdio.h"

void print(char\* a, int len)

{

int k, i, g = 0;

if (a[i] == ' ') { g = 1; }

else { g = 0; }

for (i = g; i < len;i++)

printf("%c", a[i]);

}

int main()

{

char c, buff[512];

int i = 0, k;

int m = 0, n = 0, len = 0;

while ((c = getchar()) != '!')

{

switch (c)

{

case '.': print(buff, len); printf("%c\n",c); i = 0; m++; n++; break;

case ';': print(buff, len); printf("%c\n",c); i = 0; m++; n++; break;

case '?': i = 0; m++; break;

case '\n': i = 0; break;

case ' ': i = 0; break;

default:

{

buff[i] = c;

i++;

len = i;

}

}

}

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

return 0;

}

**Вариант с динамическим выделением памяти**

Text\_v3.c

#include <stdio.h>

#include <stdlib.h>

void print\_sentence (char\* sentence, int len\_sentence)

{

int i=0;

while (sentence[i]==' ' || sentence[i]=='\n')

{

i++;

}

for (i; i < len\_sentence; i++)

{

printf("%c", sentence[i]);

}

}

int add\_char (char\* sentence, int sen\_size, int i, char c)

{

if (i > sen\_size)

{

sentence = (char\*) realloc(sentence, sen\_size\*sizeof(char)+10);

sen\_size = sen\_size+10;

}

sentence[i] = c;

return sen\_size;

}

int main()

{

int sen\_size = 10;

char c;

char\* sentence= (char\*) malloc(sen\_size\*sizeof(char));

int lenght = 0, i=0;

int sentence\_count\_before=0, sentence\_count\_after=0;

while ((c = getchar()) != '!')

{

switch (c)

{

case '\t': break;

case '.':

case ';':

{

sentence\_count\_before++;

sentence\_count\_after++;

sen\_size = add\_char(sentence, sen\_size, i, c);

i++;

sen\_size = add\_char(sentence, sen\_size, i, '\n');

i++;

print\_sentence (sentence,i);

i = 0;

break;

}

case '?':

{

sentence\_count\_before++;

i = 0;

break;

}

default:

{

sen\_size = add\_char(sentence, sen\_size, i, c);

i++;

}

}

}

printf("Количество предложений до %d и количество предложений после %d\n",sentence\_count\_before, sentence\_count\_after);

free(sentence);

return 0;

}

**Вывод**

Мы научились писать программы, используя указатели