**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 |  | Рыбин А.С. |
| Преподаватель |  | Кринкин К.В. |

Санкт-Петербург

2016

Цель:

Написание программы с использованием указателей.

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

Содержание:

#include <stdio.h>

#include <stdlib.h>

void print\_sentence (char\* buf, int lenth, int count); // Печать предложения

int main()

{

char\* buffer = (char\*)malloc(sizeof(char)\*100); // Массив для хранения ввода

int count = 0; // Длина буфера ввода

int n = 0; // Количество предложений в изначальном тексте

int m = 0; // Количество предложений в отформатированном тексте

int buf = 0;

/\* Посимвольное считывание веденного текста до знака "!" \*/

while ( (buf = getchar()) != '!')

{

/\* Количество элементов, заданное заранее, при исчерпании длины массива увеличивается в два раза \*/

int im\_count = 100;

if (count >= im\_count)

{

im\_count \*= 2;

buffer = (char\*)realloc(buffer, sizeof(char)\*im\_count);

}

buffer[count++] = (char)buf;

}

/\* Форматирование текста \*/

for (int i = 0; i < count; i++)

{

int lenth = 0; // Длина текущего предложения

if ((buffer[i] == '.') || (buffer[i] == ';'))

{

print\_sentence (buffer, lenth, i); // Печатаем очередное предложение

n++; m++; // Увеличиваем оба счетчика

lenth = 0; // Обнуляем длину текущего предложения

}

else if (buffer[i] == '?')

{

n++; // Увеличиваем только счетчик предложений до

lenth = 0; // Обнуляем длину текущего предложения

}

/\* Игнорируем табуляцию и пробелы в начале предложения и символ новой строки \*/

else if (((buffer[i] == '\t') && (lenth == 0)) || (buffer[i] == '\n') || ((buffer[i] == ' ') && (lenth == 0)));

/\* При встрече символа увеличиваем длину текущего предложения \*/

else

lenth++;

}

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

free(buffer);

system("pause");

return 0;

}

void print\_sentence (char\* buf, int lenth, int i)

/\* Печать предложения. lenth - длина текущего предложения, i - текущая позиция при проходе по массиву \*/

{

for (int g = i - lenth; g <= i; g++)

printf ("%c", buf[g]);

printf("\n");

}

**Вывод:**

Выполнив данную лабораторную работу, мы освоили и закрепили на практике написание программы с применением указателей и динамической памяти.