**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

Тема: *Использование указателей*

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Филиппенко Д.Р. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2016

Цель: написать программу, которая форматирует некоторый текст и выводит результат на консоль.

Ход работы:

1. В главной функции «main» инициализируем переменные, вызываем функцию «read» и выводим количество предложений до работы программы («after + question») и количество предложений после работы программы («after»).

int main()

{

int temp = 0;

int after = 0;

int question = 0;

while (temp == 0)

read(&temp, &after, &question);

printf("Количество предложений до %d и количество предложений после %d", (after + question), after);

return 0;

}

1. В функции «read» инициализируем массив символов и выделяем под него память с помощью malloc, предварительно подключив в препроцессоре библиотеку <stdlib.h>

char\* array = (char\*)malloc(100\*sizeof(char));

1. Воспользуемся циклом while. Он считывает символы до тех пор, пока не встретит точку, точку с запятой или вопросительный знак. В массив записываются символы, не являющиеся табуляцией, символом переноса строки или пробелом.

Если встречается символ восклицательного знака, вспомогательная переменная становится равной единице, в последующем мы этим воспользуемся.

while (((symbol = getchar()) != '.') && (symbol != '?') && (symbol != ';'))

{

if (symbol == '!')

{

\*temp = 1;

Free(array);

break;

}

if (((symbol != '\t') && (symbol != '\n') && (symbol != ' ')) || (i != 0))

array[i++] = symbol;

}

1. Если вспомогательная переменная не изменилась, используем цикл if, чтобы добавить знаки препинания в конец предложения.

Если встречается символ, равный точке или точке с запятой, то записываем его в массив и добавляем символ конца строки, затем выводим массив символов, добавляя в переменную after единицу, тем самым подсчитывая количество невопросительных предложений, то есть предложений после работы программы. Если такого символа не встречается, то добавляем единицу в переменную question, то есть подсчитываем вопросительные предложения.

if (\*temp == 0)

if ((symbol == '.') || (symbol == ';'))

{

array[i++] = symbol;

array[i] = '\0';

printf("%s\n", array);

(\*after)++;

}

else

(\*question)++;

}

**Вывод:** в ходе лабораторной работы получены навыки работы с указателями и управляющими конструкциями языка Си, которые успешно применены на практике при создании программы.