**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: *Управляющие конструкции языка Си*

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Малышенко Ю.И. |
| Преподаватель |  | Берленко Т.A. |

Санкт-Петербург

2016

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

На входпрограмме 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).

**Ход работы:**

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

int main()

{

int a = 0;

int posle = 0;//Эта переменная после компиляции без предложений с вопросами("?")

int vopros = 0;//количество "?"

while (a == 0)

read(&a, &posle, &vopros);

printf("Количество предложений до %d и количество предложений после %d", (posle + vopros), posle);

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\* massiv = (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 (((ch = getchar()) != '.') && (ch != '?') && (ch != ';'))

{

if (ch == '!')

{

\*a = 1;

break;

}

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

massiv[i++] = ch;

}

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 aft единицу, тем самым подсчитывая количество невопросительных предложений, то есть предложений после работы программы. Если такого символа не встречается, то добавляем единицу в переменную que, то есть подсчитываем вопросительные предложения.

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

{

massiv[i++] = ch;

massiv[i] = '\0';

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

++\*posle;

}

else

++\*vopros;

}

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