**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе 2**

**по дисциплине «Программирование»**

Тема: Программирование на С.

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Запевалов А.И. |
| Преподаватель |  |  |

Санкт-Петербург

2016

**Цель работы.**

Научиться работать с текстом: принимать, редактировать и выводить введенный текст.

**Ход работы.**

1. На сайте Stepik.org выполним лабораторную работу №3 из курса Программирование ЛЭТИ.
2. Рассмотрим программу.

Сначала рассмотрим следующие части функции main:

int main()

{

int i=0, m=0, n=0, j=0;

char \*buffer = (char\*)malloc(12\*sizeof(char));

while (buffer[i]!='!')

{

buffer[i] = getchar();

switch (buffer[i])

и

default:

{

i++;

if (i%10==0)

{

buffer = (char\*)realloc(buffer, (i+10)\*sizeof(char));

}

break;

}

Здесь динамически создается символьный массив, который заполняется вводимыми с клавиатуры символами. Каждый раз, когда количество символов в этом массиве становится кратным 10, его размер увеличивается на 10, что описывает блок default.

Рассмотрим действия, необходимые для разделения текста на предложения и его форматирования. Для этого используем оператор switch:

switch (buffer[i])

{

case '.':

{

while (buffer[0]=='\n'||buffer[0]=='\t'||buffer[0]==' ')

{

for (j=0; j<i; j++)

{

buffer[j]=buffer[j+1];

}

buffer[i]='\0';

i--;

}

for (j=0; j<=i; j++)

{

printf("%c", buffer[j]);

}

printf("\n");

//printf("%s\n", buffer);

memset(buffer, 0, i);

i=0;

m++;

n++;

break;

}

case ';':

{

while (buffer[0]=='\n'||buffer[0]=='\t'||buffer[0]==' ')

{

for (j=0; j<i; j++)

{

buffer[j]=buffer[j+1];

}

buffer[i]='\0';

i--;

}

for (j=0; j<=i; j++)

{

printf("%c", buffer[j]);

}

printf("\n");

//printf("%s\n", buffer);

memset(buffer, 0, i);

i=0;

m++;

n++;

break;

}

case '?':

{

memset(buffer, 0, i);

i=0;

m++;

break;

}

case '!':

{

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

break;

}

default:

{

i++;

if (i%10==0)

{

buffer = (char\*)realloc(buffer, (i+10)\*sizeof(char));

}

break;

}

}

Если в тексте попадается ‘.’ или ‘;’, срабатывает счетчик, убирается табуляция в начале предложения (происходит сдвиг массива) и печатается предложение. Если ‘?’, то срабатывает только счетчик. Каждый раз в конце этих блоков выполняется очистка массива с предложением, так как оно уже напечатано. Выход из оператора switch выполняется когда вводится восклицательный знак.

Программа полностью:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

int i=0, m=0, n=0, j=0;

char \*buffer = (char\*)malloc(12\*sizeof(char));

while (buffer[i]!='!')

{

buffer[i] = getchar();

//printf("%c", buffer[i]);

switch (buffer[i])

{

case '.':

{

while (buffer[0]=='\n'||buffer[0]=='\t'||buffer[0]==' ')

{

for (j=0; j<i; j++)

{

buffer[j]=buffer[j+1];

}

buffer[i]='\0';

i--;

}

for (j=0; j<=i; j++)

{

printf("%c", buffer[j]);

}

printf("\n");

//printf("%s\n", buffer);

memset(buffer, 0, i);

i=0;

m++;

n++;

break;

}

case ';':

{

while (buffer[0]=='\n'||buffer[0]=='\t'||buffer[0]==' ')

{

for (j=0; j<i; j++)

{

buffer[j]=buffer[j+1];

}

buffer[i]='\0';

i--;

}

for (j=0; j<=i; j++)

{

printf("%c", buffer[j]);

}

printf("\n");

//printf("%s\n", buffer);

memset(buffer, 0, i);

i=0;

m++;

n++;

break;

}

case '?':

{

memset(buffer, 0, i);

i=0;

m++;

break;

}

case '!':

{

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

break;

}

default:

{

i++;

if (i%10==0)

{

buffer = (char\*)realloc(buffer, (i+10)\*sizeof(char));

}

break;

}

}

}

return 0;

}

**Вывод.**

Мы научились принимать вводимый текст и форматировать его.