

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

Курков Д. В.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2016

Цель работы.

На практике узнать назначение указателей в языке Си, обучиться основным приемам их использования.

Задание.

Необходимо написать 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.

Ход работы:

1. Описание функции для считывания исходных данных.

```
void readtext (char **text, int *lc)
{
    char c;
    int i;
    *lc=0;
    *text = (char*)malloc(sizeof(char)); //получаем указатель
    for (i=0; (c=getchar())!='!'; i++)
    {
        /* Выделяем память на новый элемент */
        *text = (char*)realloc(*text, (i+1)*sizeof(char));
        *(*text+i) = c;
        if (c == '!' || c == ';' || c == '?')
            (*lc)++; //Считаем предложения до
    }
    *text = (char*)realloc(*text, (i+1)*sizeof(char));
    *(*text+(i++)) = '!';
    *text = (char*)realloc(*text, (i+1)*sizeof(char));
    *(*text+i) = '\0'; // Добавляем символ конца строки
}
```

2. Функции для обработки текста

1. Описание функции для удаления табуляций

```
void deltabs (char** text)
{
    int i=0, j=0;
    for (i=0, j=0; *(*text+i)!='\0'; i++, j++)
    {
        if (i!=0 && *(*text+i)==' '
            && *(*text+i-1)!='.' || *(*text+i-1)!=';' || *(*text+i-1) == '?')
            i++;
        while (*(*text+i) == '\t' || *(*text+i) == '\n')
            i++;
        *(*text+j) = *(*text+i);
    }
    *text = (char*)realloc(*text, (j+1)*sizeof(char));
    *(*text+j) = '\0';
}
```

2. Описание функции для удаления предложений, оканчивающихся знаком вопроса.

```
void delquest (char** text)
{
    int i=0, j=0;
    for (i=0, j=0; *(*text+i)!='\0' ; i++, j++)
    {
        if ((*text+i)=='?' || (*text+i)=='!')
        {
            int k;
            for(k=0; *(*text+j-(k+1))!=';' && *(*text+j-(k+1))!='.' && k<j; k++)
                ;
            j-=k;
            i++;
        }
        *(*text+j) = *(*text+i)
        if ((*text+i) == '\0')
            i--;
    }
    /* Обрезаем ненужные символы */
    *text=(char*)realloc(*text, j*sizeof(char));
}
```

3. Описание функции для добавления символов конца строки, в конце предложений.

```
char* addlines (char* str)
{
    char *text;
    int i=0, j=0;
    text = (char*)malloc(sizeof(char));
    for (i=0, j=0; *(str+i)!='\0'; i++, j++)
    {
        text = (char*)realloc(text, (j+1)*sizeof(char));
        *(text+j)=*(str+i);
        if (*(str+i) == ';' || *(str+i) == '.')
        {
            text = (char*)realloc(text, ((++j)+1)*sizeof(char));
            *(text+j)='\n';
        }
    }
    text = (char*)realloc(text, (j+1)*sizeof(char));
    *(text+j) = '\0';
    return text;
}
```

4. Функция вывода.

```
void writelines (char* str, int ln1)
{
    int ln2=0;
    for (int i=0; *(str+i)!='\0'; i++)
        if (*(str+i) == ';' || *(str+i) == '!')
            ln2++; //Считаем предложения после
    printf ("%s", str);
    printf ("Количество предложений до %d и количество предложений после
                                                    %d\n", ln1, ln2);
}
```

3. Описание функции main

```
int main ()
{
    char *text;
    int lc1;
    readtext(&text, &lc1);
    deltabs (&text);
    delquest (&text);
    text = addlines (text);
    writelines (text, lc1);
    return 0;
}
```

4. Работы программы проверена на корректность, пройдя предложенные тесты.

5. Исходный код программы помещен в репозиторий группы.

Вывод: Освоены основные приемы работы с указателями. Заданная программа написана, и успешно прошла тестирование, задание выполнено.