МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЕТ по лабораторной работе №2

по дисциплине «Программирование»

Тема: Использование указателей.

Студент гр. 0382 _____ Тихонов С.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Освоение работы с указателями и динамической памтью

Задание.

Вариант-3.

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:

- Каждое предложение должно начинаться с новой строки.
- Табуляция в начале предложения должна быть удалена.
- Все предложения, в которых есть цифры внутри слов, должны быть удалены (это не касается слов, которые начинаются/заканчиваются цифрами). Если слово начинается с цифры, но имеет и цифру в середине, удалять его все равно требуется (4a4a).
- 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**
*** Статически выделять память под текст нельзя**
*** Пробел между предложениями является разделителем, а не частью какого-то предложения**

Основные теоретические положения.

В данной лабораторной работе были использованы функции ввода и вывода gethar(), prinf() из библиотеки stdio.h. Также была использованы функции malloc(), realloc() и free() из заголовочного файла stdlib.h. Функции strcmp() и strlen() из заголовочного файла string.h. И заголовочный файл ctype.h, из которого я использовал функции isalum() isdigit() isalpha().

А так же стандартные операторы и циклы.

Выполнение работы.

Считывание текста происходит в двумерный массив номер строки в котором указывает на номер предложение, а номер столбца на номер символа в этом предложение. Считывается первое предложение до одного из символов «. ? ! ;», при помощи функции char* get_sentences(), которая возвращает одно предложение. Затем это предложение идёт в функцию char* removing_spaces которая удаляет пробел в начал предложения, если он есть. Это функция так же возвращает предложение, которое после этого попадает в функцию int normal_offer. Которая проверяет нет ли внутри слов цифр, цифра в начал слова нас устраивает, если да, то функция возвращает единицу, и данное предложение записывается в текст. Иначе, к счетчику не правильных предложений прибавляется единица, а это предложение никуда не записывается. Цикл с этими действиями повторятся пока не поступит предложение «Dragon flew away! ». После чего программа выводит обработанный текст, а так же количество предложений до обработки, и после, не учитывая последнее предложение.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

Входные данные	Данные после работы программы
Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Ut auctor augue vel tincidunt tincidunt 555. Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Ut auctor augue vel tincidunt tincidunt 555. Aliquam 555 condimentum ligula arcu, non mollis ex pell555entesque finibus. Aenean magna massa, scelerisque quis sagittis at, pharetra a lectus? Ut auctor augue vel tincidunt tincidunt 555. Dragon flew away!	Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Ut auctor augue vel tincidunt tincidunt 555. Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Ut auctor augue vel tincidunt tincidunt 555. Aenean magna massa, scelerisque quis sagittis at, pharetra a lectus? Ut auctor augue vel tincidunt tincidunt 555. Dragon flew away! Количество предложений до 7 и количество предложений после 6

Выводы.

В ходе работы была изучена работа с указателями и динамической памятью. Разработана программа, которая считывала и обрабатывает текст. В

случае нехватки памяти она перевыделялась с помощью функции realloc(). В конце программы вся память отчищается.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
char* get_sentences(){
    int sentences_size=100;
    int sentences_len=0;
    int sign;

char*sentences=malloc(sentenc
es_size*sizeof(char));
    int b=1;
    while (b==1){
        sign=getchar();
        sentences[sentences_len+
+]=sign;
        if (sign=='.' || sign=='!' ||
sign=='?' || sign==';') break;

if(sentences_len==sentences_si
ze){

sentences_size=sentences_size
*2;

sentences=realloc(sentences,
sentences_size);
        }
    }
 sentences[sentences_len]='\0';
```

```
    return sentences;
}
char*
removing_spaces(char*sentenc
es){
    int i=0;
    while (sentences[i]==' ' ||
sentences[i]=='\n' ||
sentences[i]=='\t') {
        int j;
        for (j = 0; j <
strlen(sentences) - 1; j++) {
            sentences[j] =
sentences[j + 1];
        }
        sentences[j]='\0';
    }
    return sentences;
}
int normal_offer(char
*sentences){
    int true=1;
    int false=0;
    for (int i=0;
i<strlen(sentences)-1; i++){
        if (!isalnum(sentences[i]))
{
            false=0;
        }
        else {
            if (isdigit(sentences[i]))
{
                if(!false){

if(isalpha(sentences[i+1])){
                        false=1;
                    }
                    else {
                        int j=1;
                        while
(isalnum(sentences[i+j])){
                            if
(isdigit(sentences[i+j])){
```

```
                    j++;
                }
                else{
                    true=0;
                    break;
                }
            }
            i+=j;
        }
    }
    else{
        if
(isalnum(sentences[i+1])){
            true=0;
            break;
        }
    }
}
else{
    false=1;
}
}
}
return true;
}
int main(){
    int text_len=0;
    int wrong_sentences=0;
    char* end_text= "Dragon
flew away!";
    int text_size =60;

char**text=malloc(text_size*si
zeof(char*));
    char*sentences;
    int a=1;
    while (a==1){

sentences=get_sentences();
        sentences=
removing_spaces(sentences);
        if
(normal_offer(sentences)){
```

```
        text[text_len+
+]=sentences;
        }
        else wrong_sentences
+=1;
        if (text_len==text_size){
            text_size=text_size*2;
            text=realloc(text,
text_size*sizeof(char*));
        }
        if(!strcmp(sentences,
end_text)) break;
    }
    for (int i=0; i<text_len; i++){
        puts (text[i]);
        free(text[i]);
    }
    free(text);
    printf("Количество
предложений до %d и
количество предложений
после %d\n",
text_len+wrong_sentences-1,
text_len-1);
    return 0;
}
```