# 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: Указатели и массивы

|  |  |  |
| --- | --- | --- |
| Студент гр. 1384 |  | Усачева Д.В. |
| Преподаватель |  | Жангиров Т. Р. |

Санкт-Петербург 2021

# Цель работы.

Целью работы является освоения использования массивов в языке C, аккуратная работа с указателями, управлением памяти.

# Задание.

Вариант №2

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, которые заканчиваются на '?' должны быть удалены.
* 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**

# Выполнение работы.

Для удобства работы предложение было определено как отдельная

структура. Она содержит два поля: str (массив символов, представляющий

это предложение), len (текущее количество символов в массиве. Аналогично был определен текст.

Для считывания текста были реализованы функции readSentence и readText которые считывает входящий текст по предложениям в динамический массив, а также находит длину этого массива.

Для удаления пробелов и табуляции до начала предложения была реализована функция del\_tab, которая меняет исходный текст и соответственно его длину.

Для удаления предложений, имеющих «?» была реализована функция del\_qwe. В функции производится подсчет количества предложений в новом тексте (уменьшает их количество, если удаляется предложение). Функция ищет предложения, в которых символом конца является «?» такие предложения удаляются. В теле функции main создается переменная n, хранящая длину введенного массив символов. В переменной text хранится введенный массив символов. С помощью функции free очищается вся выделенная память (массив введенных символов, массив предложений, массив символов (сами предложения).

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

# Выводы.

Была освоено использование массивов в языке C, работа с указателями

и управление памятью.

# ТЕСТИРОВАНИЕ

Таблица 2.1 - Пример тестовых случаев

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
| 1. | Nulla facilisi. Class aptenT taciti sociosqu ad litora torquent per cOnubia nostra, per inceptos himenaeos. 40 Nu555llarutrum feugiat felis a pharetra. Sed finibus magna et mauris elementum tempus? Integer at quam et erat iaculis iaculishendrerit a te4llus? Donec at nunc ac mauris suscipit venenatis. Sed finibus magna et mauris elementum tempus? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi co7ndimentum 555 ex justo, nec pharetra mauris vestibulum a. Suspendisse quis mi neque7. 1 Vivamus eu nibh rhoncus, da456pibus ex non, sodales mi. Donec accumsan convallis ipsum vitae lacinia. Donec accumsan convallis ipsum vitae lacinia. Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Nulla facilisi. Lorem ipsum dolor sit amet, onsectetur adipiscing elit. Dragon flew away! | Nulla facilisi. Class aptenT taciti sociosqu ad litora torquent per cOnubia nostra, per inceptos himenaeos. 40 Nu555lla rutrum feugiat felis a pharetra. Sed finibus magna et mauris elementum tempus? Integer at quam et erat iaculis iaculis hendrerit a te4llus? Donec at nunc ac mauris suscipit venenatis. Sed finibus magna et mauris elementum tempus? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse quis mi neque7. 1 Vivamus eu nibh rhoncus, da456pibus ex non, sodales mi. Donec accumsan convallisipsum vitae lacinia. Donec accumsan convallis ipsum vitae lacinia. Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Dragon flew away! Количество предложений до 16 и количество предложений после 15 | Реализованная программа корректно работает: обрабатывает введенный текст и выводит количество предложений до обработка и после (без учета терминального). |

# ПРИЛОЖЕНИЕ 1

# ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include <ctype.h>

#define MEM\_STPEP 20

//предложение

typedef struct Sentence {

    char \*str;

    int len;//!= strlen(str) размер буффера, не длина

} Sentence;

//текст

typedef struct Text {

    struct Sentence \*\*sents;

    int size;

    int len;

} Text;

Sentence \*readSentence() {

    int size = MEM\_STPEP;

    char \*buf = malloc(size \* sizeof(char));

    char temp;

    int n = 0;

    do {

        if (n >= size - 2) {

            char \*t = realloc(buf, size\*2);

            if (t) {

                size \*=2;

                buf = t;

            }

        }

        temp = getchar();

        buf[n] = temp;

        n++;

    } while (temp != ';' && temp != '.' && temp != '?' && temp != '!');

    buf[n] = '\0';

    Sentence \*sentence = malloc(sizeof(Sentence));

    sentence->str = buf;

    sentence->len = n;

    return sentence;

}

Text \*readText() {

    int size = MEM\_STPEP;

    Sentence \*\*text = malloc(size \* sizeof(Sentence \*));

    char end='\0';

    int n = 0;

    Sentence \*temp;

    do {

        if (n >= size - 2) {

            Sentence \*\*t = realloc(text, size\*2);

            if (t) {

                text=t;

                size \*= 2;

            }

        }

        temp = readSentence();

        text[n] = temp;

        n++;

    }

    while ((temp->str[(temp->len)-1]) !='!');

    Text \*txt= malloc(sizeof(Text));

    txt->len = n;

    txt->sents = text;

    txt->size = size;

    return txt;

}

void del\_tab(Text txt,int n){

    for (int i=0; i<n;i++){

        if (((((txt.sents)[i]->str)[0])=='\t' ) || ((((txt.sents)[i]->str)[0])==' ' )){

            char \*old\_in=&txt.sents[i]->str[0];

            char \*new\_in=&(txt.sents[i]->str[0])+1;

            int l=strlen(txt.sents[i]->str);

            memmove(old\_in,new\_in,l);

        }

    }

}

int del\_qwe(Text txt, int n){

    int k=n;

    for (int i=0; i<n;i++){

        int end\_s=strlen(txt.sents[i]->str);

        if (txt.sents[i]->str[end\_s-1]=='?'){

            k--;

            char \*old\_in = &txt.sents[i]->str[0];

            char \*new\_in = &txt.sents[i]->str[end\_s];

            memmove(old\_in,new\_in,end\_s);

        }

    }

    return k;

}

int main() {

    Text \*text = readText();

    Text t=\*text;

    int n=text->len;

    del\_tab(t,n);

    int k=del\_qwe(t,n);

    puts("------------------------------");

    for (int i=0; i<n; i++)

        printf("%s\n",(text->sents)[i]->str);

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

    return 0;

}