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: Использование указателей

Студент гр. 0383: Шквиря Е.В.

Преподаватели: Чайка К. В.
Жангиров Т. Р.

Санкт-Петербург
2020

Цель работы.

Изучить устройство указателей и их использование в языке Си.

Задание.

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 больше одной заглавной буквы, 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.

Порядок выполнения поставленной задачи программой:

1) Сначала создаётся указатель *text*, чтобы записать в него адрес строки, хранящей текст. Функция *enter_text* резервирует память для вводимых данных и записывает туда текст.

2) Функция *enter_text* считывает текст по алгоритму:

1. Если встретился символ, являющийся концом предложения (. ; ?), то увеличивается счётчик предложений и фиксируется окончание предложения в переменной *wasAlpha*;
2. Если встретился символ, который не является табуляцией, переносом строки или пробелом, а также не обработан первым условием, то он является частью предложения, в дальнейшем фиксируется факт, что оно началось, иначе этот символ нужно пропустить;
3. Если недостаточно места, чтобы записать символ, то функцией *realloc* увеличивается память, выделенная под текст, по формуле 1.7 * *maxSize* (константа 1.7 взята для того, чтобы не выделялось сильно много памяти). После этого каждый символ записывается, а после прочтения текста в конце ставится \0 и возвращается указатель на строку.

3) Обработка текста производится по алгоритму:

1. Пока не встретился признак конца предложения, то считается количество символов в верхнем регистре, также высчитывается длина предложения.
2. Если количество букв в верхнем регистре меньше 2, то это предложение сохраняется. Удаление неподходящих предложений

происходит путём записывания подходящих поверх них. Таким образом, по новому индексу записывается символ из старого индекса $(newInd \le oldInd)$. Если предложение подошло, то увеличивается счётчик новых (актуальных) предложений.

3. После обработки всех предложений ставится \0 в конце.

4) Вывод осуществляется до символа \0, каждое предложение начинается с новой строки.

Используемые переменные:

cntSnt — количество предложений при считывании;

cntActualSnt — количество предложений для вывода;

text — указатель на текст, вводимый пользователем;

oldInd — индекс в старой версии текста;

newInd — индекс в новой версии текста;

cntCapit — количество символов в предложении в верхнем регистре;

size — количество символов в предложении;

В функции enter_text:

c — символ, введённый пользователем;

maxSize — размер выделенной памяти;

size — количество символов в предожении;

text — указатель на текст, который в дальнейшем будет возвращён как результат работы функции;

wasAlpha — показатель, началось ли предложение;

temp — временная переменная для текста;

Дополнительно используется макроопределение:

INCREASE — константа для увеличения размера предложения.

Используемые фукции:

enter_text — ввод текста и формирование строки, в которой он содержится.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Nu555llam auctor vehicula dui, …… sed ultricies nulla. Dragon flew away!	Nu555llam auctor vehicula dui, quis lobortis nibh. …… Dragon flew away! Количество предложений до 185 и количество предложений после 167	Верный ответ
2.	Hello, how are you? All right. What is it? Is it a dragon? No. It is a picture. Dragon flew away!	Hello, how are you? All right. What is it? Is it a dragon? No. It is a picture. Dragon flew away! Количество предложений до 6 и количество предложений после 6	Верный ответ
3.	AabbaaA? abbacaba. aa; AA; bb? AAAAAAAAAAAAAAA? abbab. Dragon flew away!	abbacaba. aa; bb? abbab. Dragon flew away! Количество предложений до 7 и количество предложений после 4	Верный ответ

Выводы.

Было изучено устройство указателей, способы их использования и работа со строками в языке Си.

Разработана программа, получающая от пользователя текст и обрабатывающая его, удаляя предложения и убирая лишние разделяющие символы. Для реализации использовались функции для работы с памятью, такие как *malloc* и *realloc*, передача указателя в функцию и возвращение указателя как результат выполнения функции.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.c

```
#include <stdio.h>
#include <malloc.h>
#include <ctype.h>
#include <locale.h>


#define INCREASE 1.7


char* enter_text(int *cntSnt)
{
	char c;
	int maxSize = 10;
	int size = 0;
	char *text = (char *) malloc(maxSize * sizeof(char));
	int wasAlpha = 0;
	while ((c = getchar()) != EOF)
	{
		if (c == '.' || c == ';' || c == '?')
		{
			(*cntSnt)++;
			wasAlpha = 0;
		}
		else if(c != '\t' && c != ' ' && c != '\n')
			wasAlpha = 1;
		else if(!wasAlpha || c == '\n')
			continue;
		if (size + 1 == maxSize)
		{
			char* temp = (char*)realloc(text,(int)(maxSize * INCREASE) *
sizeof(char));
			maxSize *= INCREASE;
			if(temp)
				text = temp;
			else
```

```
                        return NULL;
                }
                text[size++] = c;
        }
        text[size] = '\0';
        return text;
}

int main()
{
        setlocale(LC_ALL, "");
        //считывание текста
        int cntSnt = 0;
        int cntActualSnt = 0;
        char *text;
        text = enter_text(&cntSnt);
        if(!text)
                return -1;

        //обработка текста
        int oldInd = 0, newInd = 0;
        int cntCapit = 0;
        int size = 0;
        while(text[oldInd] != '\0')
        {
                while (text[oldInd] != '.' && text[oldInd] != ';' && text[oldInd] != '?' &&
text[oldInd] != '!')
                {
                        if(isupper(text[oldInd]))
                                cntCapit += 1;
                        oldInd++;
                        size++;
                }
                if(cntCapit < 2)
                {
                        for (int i = 0; i < size+1; ++i)//+1 из-за конечного символа(. ? ;)
                                text[newInd++] = text[oldInd - size + i];
                        cntActualSnt++;
                }
```

```
                cntCapit = 0;
                size = 0;
                oldInd++;
        }
        text[newInd] = '\0';
        //вывод
        newInd = 0;
        while(text[newInd] != '\0')
        {
                printf("%c", text[newInd]);
                if(text[newInd] == '.' || text[newInd] == ';' || text[newInd] == '?')
                        puts("");
                newInd++;
        }
        cntActualSnt--;
        printf("\nКоличество предложений до %d и количество предложений после %d",
cntSnt, cntActualSnt);


        free(text);
        return 0;
}
```