

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

Студентка гр. 1304

Виноградова М.О.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Написать программу в соответствии с условием задачи.

Задание.

Вариант 1

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 цифра 7 (в любом месте, в том числе внутри слова), 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).

Выполнение работы.

temp - переменная в которой храниться считанный символ. **sent**-двумерный массив в котором будут храниться предложения. **num_sent**, **num_el** счетчики для номеров предложений и символов в предложении соответственно. Предложение считывается в цикле пока на вход не поступит терминальное предложение "Dragon flew away!" записанное в переменную **end**. Символы считываются до знака разделителя предложений (‘;’ ‘?’ ‘.’) и записываются в массив **sent[num_sent]**. Если была найдена семерка (для отслеживания создана переменная **find_seven**), то переменная **num_sent** не изменяется, а записанные ранее символы обнуляются. Если семерка не была найдена, то происходит запись предложения в массив. Начальное количество предложений храниться в переменной **len**. Количество предложений после обработки в соответствии с условием задачи будет равно **num_sent**.

Память выделяется динамически с помощью функции **malloc()**, если памяти становится недостаточно, то она расширяется с помощью функции **realloc()**. Освобождение памяти осуществляется с помощью функции **free()**.

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.	Dragon.Dragon flew away!	Dragon. Dragon flew away! Количество предложений до 1 и количество пред ложений после 1	Программа работает корректно
2.	Dragon. Second dragon? Dragon flew away!	Dragon. Second dragon? Dragon flew away! Количество предложений до 2 и количество предл ожений после 2	Программа работает корректно

Вывод.

В соответствии с условием задачи была реализована программа.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>
#include<string.h>
#include<stdlib.h>
#define N 200
#define M 200

int main(void) {

    char temp='q';
    char **sent;
    int size_sent=N;
    int size_word=M;
    char c;
    char* more;
    char** more_more;
    sent = malloc(size_sent * sizeof(char*));
        for(int i=0;i<size_sent;i++)
sent[i] = malloc(size_word * sizeof(char));
```

```

int len=0;
int num_sent=0,num_el=0;
int find_seven=0;
char *end="Dragon flew away!";
while(strcmp(end,sent[num_sent])!=0){

    temp=getchar();

    if(temp!='?' && temp!='.' && temp!=';'){

        if(size_word-num_el<=3){
            size_word+=M;
            more=realloc(sent[num_sent],size_word);
            sent[num_sent]=more;
        }

        if(temp=='7')find_seven=1;

        sent[num_sent][num_el]=temp;
        num_el++;

    }else{

        if(size_sent-num_sent<=3){
            size_sent+=N;
            more_more=realloc(sent,size_sent);
            sent=more_more;
        }

        sent[num_sent][num_el]=temp;
        num_el++;
        sent[num_sent][num_el]='\0';
        scanf("%c",&c);
        len++;
        if(find_seven==0){
            num_sent++;
        }else{
            for(int i =0;i<num_el;i++)
                sent[num_sent][i]=0;
        }
        num_el=0;
        find_seven=0;

    }

}

}

```

```

    for(int i=0;i<num_sent;i++){
        printf("%s\n",sent[i]);
    }
    printf("Dragon flew away!\n");
    printf("Количество предложений до %d и количество предложений после %d",len,num_sent);

    for(int i=0;i<N;i++)
        free(sent[i]);
    free(sent);

    return 0;
}

```