

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: Сборка программ в Си

Студент гр. 1304

Арчибасов Е.О.

Преподаватель

Чайка К.В.

Санкт-Петербург

2021

Цель работы.

Освоение работы с указателями и динамической памятью.

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 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).

*** 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.

- `malloc (void* malloc (size_t size))` – выделяет блок из size байт и возвращает указатель на начало этого блока
- `realloc (void* realloc (void* p, size_t size))` – изменяет размер ранее выделенной области памяти на которую ссылается указатель p. Возвращает указатель на область памяти, измененного размера.
- `free (void free (void* p))` – высвобождает выделенную ранее память.

Ход работы.

Для написания программы нам потребуется подключить следующие библиотеки: `<stdio.h>`, `<stdlib.h>` и `<string.h>`. Так же в начале программы объявим два макроса `#define size 5` и `#define endStr "Dragon flew away!"`

Первой функцией является `enterTextArr()`. Целочисленная переменная `nStr = 0`, отвечающая за количество предложений в тексте, двумерный массив `textArr`, в который мы запишем текст и для которого выделим динамическую память `malloc(size * sizeof(char*))`, символьная переменная `symb`. С помощью цикла `do {} while`, пока проверка с помощью функции `strcmp(textArr[nStr-1], endStr)` не покажет, что последнее предложение идентично предложению `endStr` в цикле происходит считывание текста. Если выделенной раньше для двумерного массива памяти не хватает, происходит увеличение. Для предложения `textArr[nStr]` динамически выделяем память. Объявим целочисленную переменную `nSymb = 0`, отвечающая за количество символов в предложении. Считаем пробел в начале предложения, если он есть. Таким образом добьемся корректного вывода результата текста. В этом же цикле начинает работать следующий цикл до тех пор пока введенный символ не будет одним из нужных нам (‘.’, ‘;’, ‘?’, ‘!’). После завершения работы внутреннего цикла, во внешнем происходит изменение памяти под конкретный объем равный количеству символов + 1 (учитывается обязательный символ строк), последнему значению присваивается этот символ ‘\0’. Функция возвращает двумерный массив `textArr`.

Функция `nStrFirst (char **textArr)`. Функция принимает на вход двумерный массив – введенный текст и считает количество предложений в нем. Объявляется целочисленная переменная `nStr = 0`. По завершению работы цикла возвращается значение равное количеству предложений в исходном тексте `nStr`.

Функция `editTextArr (char **textArr, int nStrFirst)`, принимает на вход количество предложений в исходном тексте и сам текст. Объявляется целочисленная переменная `nStr = 0`, которая используется в цикле `while()`,

который работает до тех пор пока количество этих предложений не равно конечному $nStr \neq nStrFirst$. В цикле объявляется целочисленная переменная $k = 0$, которая отвечает за нахождение символа числа 7 внутри предложений текста. С помощью цикла *for* пробегаем по всем значениям массива. При выходе из цикла *for* если значение переменной $k > 0$, то память, выделенная для этого предложения, очищается и происходит сдвиг предложений. По завершению работы цикла программа возвращает исправленный в соответствии с заданием текст `textArr`.

В главной функции *main()* объявляем двумерный массив `textArr`. В целочисленную переменную `nFirst` запишем результат работы функции `nStrFirst()`. Далее `textArr` обрабатывается функцией `editTextArr()`. В целочисленную переменную `nLast` запишем результат работы функции `nStrFirst()`, на вход поступает уже измененный текст. С помощью цикла *for* выведем отредактированный текст с применением функции *printf()*. С помощью функции *printf()* выводятся переменные `nFirst` и `nLast` – количество предложений до и количество предложений после соответственно. С помощью цикла *for* происходит освобождение динамически выделенной памяти на каждое предложение `free(textArr[i])`. В конце очищается память выделенная на весь текст динамически `textArr(txt)`.

Разработанный программный код см. в приложении 1.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Donec accumsan convallis ipsum vitae lacinia. 1 Vivamus eu nibh rhoncus, da456ribus ex non, sodales	Donec accumsan convallis ipsum vitae lacinia. 1 Vivamus eu nibh rhoncus,	Верно

	mi. Aliquam 555 condimentum ligula arcu, non mollis ex pell555entesque finibus. Nu555llam auctor vehicula dui, quis lobortis nibh. Dragon flew away!	da456pibus ex non, sodales mi. Aliquam 555 condimentum ligula arcu, non mollis ex pell555entesque finibus. Nu555llam auctor vehicula dui, quis lobortis nibh. Dragon flew away! Количество предложений до 4 и количество предложений после 4	
2.	saldsnvljndaslvjnalvaldvlak vmas;vlaksdvnljdnvlnanv.as vjansjlvnalskvnlasnv;Drago n flew away!	saldsnvljndaslvjnalvaldvlak vmas; vlaksdvnljdnvlnanv. asvjansjlvnalskvnlasnv; Dragon flew away! Количество предложений до 3 и количество предложений после 3	Верно
3.	Lorem ipsum dolor sit amet, consectetur adipiscing elit; Aenean sem ligula, laoreet ac sodales a, congue euismod neque; Donec at nunc ac mauris suscipit venenatis. Sed finibus magna et mauris elementum tempus? Maecenas 555 posuere velit efficitur, egestas nunc quis, dictum purus? Aenean magna massa, scelerisque quis sagittis at, pharetra a lectus? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla facilisi. Nu555llam auctor vehicula dui, quis lobortis nibh. Donec accumsan convallis ipsum vitae	Lorem ipsum dolor sit amet, consectetur adipiscing elit; Aenean sem ligula, laoreet ac sodales a, congue euismod neque; Donec at nunc ac mauris suscipit venenatis. Sed finibus magna et mauris elementum tempus? Maecenas 555 posuere velit efficitur, egestas nunc quis, dictum purus? Aenean magna massa, scelerisque quis sagittis at, pharetra a lectus? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla facilisi. Nu555llam auctor	Верно

<p> lacinia. Vivamus sit amet viverra arcu, sed ultricies nulla. Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Cras eget felis nibh? Suspendisse quis mi neque7. Aenean sem ligula, laoreet ac sodales a, congue euismod neque; Aenean sem ligula, laoreet ac sodales a, congue euismod neque; Donec accumsan convallis ipsum vitae lacinia. 1 Vivamus eu nibh rhoncus, da456pibus ex non, sodales mi. Aliquam 555 condimentum ligula arcu, non mollis ex pell555entesque finibus. Nu555llam auctor vehicula dui, quis lobortis nibh. Dragon flew away! </p>	<p> vehicula dui, quis lobortis nibh. Donec accumsan convallis ipsum vitae lacinia. Vivamus sit amet viverra arcu, sed ultricies nulla. Fusce finibus sapien magna, quis scelerisque ex sodales tristique. Cras eget felis nibh? Aenean sem ligula, laoreet ac sodales a, congue euismod neque; Aenean sem ligula, laoreet ac sodales a, congue euismod neque; Donec accumsan convallis ipsum vitae lacinia. 1 Vivamus eu nibh rhoncus, da456pibus ex non, sodales mi. Aliquam 555 condimentum ligula arcu, non mollis ex pell555entesque finibus. Nu555llam auctor vehicula dui, quis lobortis nibh. Dragon flew away! Количество предложений до 20 и количество предложений после 19 </p>
---	--

Вывод: В ходе выполнения лабораторной работы мы научились работать с указателями в языке си и многомерными массивами

Приложение 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define size 5
#define endStr "Dragon flew away!"

char **enterTextArr(){
    char **textArr= malloc(size * sizeof(char*));
    char symb;
    int nStr= 0;

    do{
        textArr= realloc(textArr, (nStr + 1) * sizeof(char*));
        textArr[nStr]= malloc(size * sizeof(char));

        int nSymb= 0;

        scanf(" ");
        do{
            scanf("%c", &symb);
            textArr[nStr]= realloc(textArr[nStr], (nSymb + 1) * sizeof(char));
            textArr[nStr][nSymb]= symb;
            nSymb+= 1;
        } while((symb != '.') && (symb != ';') && (symb != '?') && (symb != '!'));

        textArr[nStr]= realloc (textArr[nStr], (nSymb + 1) * sizeof(char));
        textArr[nStr][nSymb]= '\0';
        nStr+= 1;
    } while(strcmp(textArr[nStr-1], endStr) != 0);

    return textArr;
}

int nStrFirst(char **textArr){
    int nStr= 0;

    while (strcmp(textArr[nStr], endStr) != 0){
```

```

        nStr+=1;
    }

    return nStr;
}

char **editTextArr(char **textArr, int nStrFirst){
    int nStr= 0;
    int k= 0;

    while ( nStr != nStrFirst){
        for (int i= 0; i < strlen(textArr[nStr]); i++ ){
            if (textArr[nStr][i]== '7'){
                k+= 1;
            }
        }

        if (k > 0){
            free (textArr[nStr]);
            for (int i= nStr ; i < nStrFirst - 1; i++)
                textArr[i]= textArr[i+1];
        } else{
            nStr+= 1;
        }
        k= 0;
    }

    return textArr;
}

int main(){
    char **textArr= enterTextArr();

    int nFirst= nStrFirst(textArr);
    textArr= editTextArr(textArr, nFirst+1);

    int nLast= nStrFirst(textArr);

    for(int i= 0; i < nLast; i++)
        printf ("%s\n", textArr[i]);

    printf ("%s\n", endStr);
    printf ("Количество предложений до %d и количество предложений после %d\n", nFirst,
nLast);
}

```



```
for(int i= 0; i < nLast+1; i++){  
    free(textArr[i]);  
}  
  
free(textArr);  
return 0;  
}
```