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. 1304 _____ Кривоченко Д.И.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2021

Цель работы.

Научиться работать с указателями, динамической памятью, текстом 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 (\t, ' ') в начале предложения должна быть удалена.

Все предложения, в которых есть число 555, 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.

1. malloc (int *size). Выделение динамической памяти размера size байт.
2. realloc (char *temp,int size). Увелечение или уменьшение памяти массива temp на size байт.
3. Оператор * - разыменование элемента.
4. Оператор & - получение адреса.
5. free(temp) - освобождение памяти массива элементов.

Выполнение работы.

Вся работа выполняется в главной функции main(). Были заведены переменные endl (для прерывания считывания текста), sentTern (для окончания считывания предложения), termChar (дабы не выводить предложения с символом termChar), а также флаги flagSent (для прерывания считывания предложения), flagText (для прерывания считывания текста). Также понадобились счетчик предложений sentCount и счетчик слов в предложении (charCount), а также счетчик для вывода ответа (кол-во предложений до и после) countansw.

Память для предложений выделялась построчно по мере надобности с помощью realloc. Память для букв внутри предложений выделялась пластом размера strLen.

После вывода предложений, память освобождалась с помощью функции free.

Тестирование.

Здесь результаты тестирования, которые помещаются на одну страницу.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	7asd.Dragon flew away!	Dragon flew away! Количество предложений до 1,после 0.	Удалено предложение, содержащее 7.

2.	Bazinga. Dragon flew away!	Bazinga. Dragon flew away! Количество предложений до 1,после 1.	Ничего не удалено
3.	I was there! 7 wasn’t .Dragon flew away!	I was there! Dragon flew away! Количество предложений до 1,после 1.	Удалено предложение, содержащее 7.

Выводы.

Изучена работа с указателями, динамической памятью, текстом.

Была написана программа на языке си, в которой была совершена работа с динамической памятью.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Сначала указываем имя файла, в котором код лежит в репозитории:

Название файла: main.c

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define strLen 1000


int
main ()
{

  char c;
  char *endl = "Dragon flew away!";
  char *sentTern = ".;!?";
  char **text = malloc (sizeof (char *));
  char termChar = '7';
  int flagSent = 1;
  int flagText = 1;
  int sentCount = 0;
  int charCount = 0;
  int countansw = 0;
  while (flagText)
    {
      charCount = 0;
      text = realloc (text, sizeof (char *) * (sentCount + 1));
      text[sentCount] = malloc (sizeof (char) * strLen);
      c = getchar ();
      text[sentCount][charCount] = c;


      do
      {
        charCount++;
        c = getchar ();
        text[sentCount][charCount] = c;
        flagSent = (strchr (sentTern, c) == NULL);

      }
      while (flagSent == 1);

      text[sentCount][charCount + 1] = '\0';
      c = getchar ();
      flagText = (strstr (text[sentCount], endl) == NULL);
      sentCount++;

    }

  for (int i = 0; i < sentCount; i++)
    {
```

```
            if (strchr (text[i], termChar) == NULL)
            {
              printf ("%s\n", text[i]);
              countansw++;
            }
            free (text[i]);
          }
        free (text);
        printf("Количество предложений до %d и количество предложений
после %d\n", sentCount-1 ,countansw-1);
        return 0;
      }
```