

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Программирование»**  
**Тема: «Обзор стандартной библиотеки»**

Студент гр. 9303

\_\_\_\_\_

Максимов Е.А.

Преподаватель

\_\_\_\_\_

Чайка К.В.

Санкт-Петербург

2019

### **Цель работы.**

Изучить основные функции стандартной библиотеки языка программирования C.

### **Задание.**

Вариант лабораторной работы №1.

Напишите программу, на вход которой подается текст на английском языке (длина текста не превышает 1000 символов) и слово *str* (длина слова не превышает 30 знаков). Слова в тексте разделены пробелами или точкой. Программа должна вывести строку «exists», если *str* в тексте есть и «doesn't exist» в противном случае.

Программа должна реализовать следующий алгоритм:

- 1) разбить текст на слова, используя функции стандартной библиотеки;
- 2) отсортировать слова, используя алгоритм быстрой сортировки (см. функции стандартной библиотеки);
- 3) определить, присутствует ли в тексте *str*, используя алгоритм двоичного поиска (для реализации алгоритма двоичного поиска используйте функцию стандартной библиотеки);
- 4) вывести строку «exists», если *str* в тексте есть и «doesn't exist» в противном случае.

### **Основные теоретические положения.**

Стандартная библиотека языка C - часть стандарта ANSI C, посвящённая заголовочным файлам и библиотечным подпрограммам. Является описанием реализации общих операций, таких как обработка ввода-вывода и строк.

1) *stdlib.h* – заголовочный файл стандартной библиотеки языка C, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие. Включает в себя следующие функции:

- *malloc*, *realloc*, *calloc*, *free* – функции для работы с динамической памятью;

- *qsort* – функция быстрой сортировки массива, разработанная С. Р. Хори.

Прототип функции выглядит следующим образом:

*void qsort(void \*base, size\_t num, size\_t size, int (\*compare) (const void \*, const void \*));*, где:

*void \*base* – указатель на массив, который нужно сортировать;

*size\_t num* – количество сортируемых элементов массива;

*size\_t size* – размер каждого сортируемого элемента в байтах;

*int (\*compare) (const void \*, const void \*)* – функция-компаратор, сигнатура которой выглядит следующим образом:

*int compare (const void \*, const void \*);*, где:

*const void \** – сравниваемые элементы.

Функция должна возвращать значения меньше 0, равное 0 или больше 0, если первый элемент младше, равен или больше второго соответственно.

- *bsearch* – функция, выполняющая двоичный поиск в отсортированном массиве. Прототип функции:

*void \*bsearch(const void \*key, const void \*base, size\_t num, size\_t size, int (\*compare)(const void\*, const void\*));*, где:

*const void \*key* – указатель на искомый элемент в массиве;

*const void \*base* – указатель на отсортированный массив;

*size\_t num* – количество элементов отсортированного массива;

*size\_t size* – размер каждого элемента отсортированного массива в байтах;

*int (\*compare)(const void\*, const void\*)* – функция-компаратор, аналогичная функции-компаратору *qsort*.

Функция возвращает указатель на элемент массива, совпадающий с искомым элементом *key*, или *NULL*, если он отсутствует.

Достоинство функции заключается в том, что при множественном вызове *bsearch* функция работает быстрее, чем тривиальное сравнение каждого элемента с искомым.

2) *string.h* – заголовочный файл стандартной библиотеки языка Си, содержащий функции для работы с нуль-терминированными строками

(заканчивающимся символом «\0» и различными функциями работы с памятью. Включает в себя следующие функции:

- *strtok* – функция, разбивающая строку на лексемы (подстроки), разделенные символами-разделителями. Прототип функции:

*char \*strtok(char \*string, const char \*delim);* , где:

*char \*string* – строка, которую необходимо разделить;

*const char \*delim* – строка, содержащая разделители подстрок.

Функция находит первое вхождение одного из разделителей и заменяет его символом «\0», и возвращает указатель на данную подстроку. Для повторного вызова функции в качестве первого аргумента необходимо указать *NULL*. В случае, если подстрок больше не оказалось, функция возвращает *NULL*.

### **Выполнение работы.**

В программе использовались следующие переменные:

1. Целого типа (integer):

- a. *counter* – количество подстрок, разделённых разделителями слов или предложений.

2. Указатели (char\*):

- a. *text* – строка, представляющая собой текст на английском языке, в котором производится поиск строки *str*;
- b. *str* – искомая подстрока;
- c. *delim* – строка, содержащая разделители слов или предложений.
- d. *p* – указатель, участвующий в цикле разбиения текста на подстроки.

3. Указатели (char\*\*):

- a. *word* – массив указателей на подстроки.

В программе реализованы следующие функции:

1. Функция `int qsortcmp(const void *ptr1, const void *ptr2)` принимает на вход два указателя на строки и производит лексикографическое сравнение строк. Используется в вызове функции `qsort`.
2. Функция `int bsearchcmp(const void *ptr1, const void *ptr2)` принимает на вход два указателя на строки, один из которых является искомой строкой, и производит лексикографическое сравнение строк. Используется в вызове функции `bsearch`.

Разработанный программный код см. в приложении А.

### Результаты тестирования.

№ п/п	Входные данные	Выходные данные	Комментарии
1.	One of the most alarming forms of air pollution is acid rain. air	exists	Тест пройден
2.	Java is a general-purpose computer programming language that is concurrent class-based object-oriented and specifically designed to have as few implementation dependencies as possible. java	doesn't exist	Тест пройден
3.	Marco Polo is famous for his journeys across Asia. He was one of the first Europeans to travel in Mongolia and China. He wrote a famous book called The Travels. was	exists	Тест пройден
4.	The universe is enormous, so the chances of us being the only living creatures are small. chance	doesn't exist	Тест пройден
5.	The universe is enormous, so the chances of us being the only living creatures are small. universe	exists	Тест пройден

### Выводы.

В ходе лабораторной работы были изучены основные функции стандартной библиотеки языка программирования C. В результате

лабораторной работы была разработана программа для поиска слова в тексте, который предварительно разбивается на слова. Для обработки текста использовались функции стандартных библиотек *stdlib.h* и *string.h*.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

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

int qsortcmp(const void *ptr1, const void *ptr2){
    return strcmp(*(char**)ptr1, *(char**)ptr2);
}

int bsearchcmp(const void* ptr1, const void* ptr2){
    return strcmp((char*)ptr1, *(char**)ptr2);
}

int main(){
    char text[1001], str[31];
    char delim[] = " .";
    fgets(text, 1000, stdin);
    fgets(str, 30, stdin);

    char **word = (char**)malloc(sizeof(char*));
    char* p = strtok(text, delim);
    int counter;

    for(counter=0; p; counter++){
        word[counter]=p;
        p = strtok(NULL, delim);
        word = realloc(word, (counter+2)*sizeof(char*));
    }
    counter--;

    qsort(word, counter, sizeof(char*), qsortcmp);
    if(bsearch(str, word, counter, sizeof(char*),
bsearchcmp)==NULL)
        printf("doesn't exist");
    else printf("exists");

    return 0;
}
```