

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Программирование»**  
**Тема: Обзор стандартной библиотеки**

Студентка гр. 0382

Здобнова К.Д.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2021

## **Цель работы.**

Изучить функции стандартной библиотеки Си.

## **Задание.**

Вариант 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:

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

## **Основные теоритические сведения.**

```
void qsort (void* base, size_t num, size_t size,  
            int (*compare)(const void*,const void*));
```

Функция принимает указатель на начальный элемент массива, количество элементов и размер одного элемента, а также указатель на функцию для сравнения двух элементов.

Так как тип элементов может быть любым, то и указатель на первый элемент массива имеет тип `void`. Это позволяет, зная адрес первого элемента и размер каждого элемента вычислить адрес любого элемента массива в памяти и обратиться к нему. Остается только сравнить 2 элемента имея 2 указателя на них. Это выполняет функция `compare`, указатель на которую передается функции `qsort` в качестве одного из параметров.

Функция `compare` принимает 2 указателя типа `void`, но в своей реализации может привести их к конкретному типу (так как её реализация остается за программистом, он точно знает элементы какого типа он сортирует) и сравнивает их. Результат сравнения определяется знаков возвращаемого функций `qsort` числа.

```
void* bsearch (const void* key, const void* base, size_t num, size_t size, int  
               (*compar)(const void*,const void*));
```

Выполняет поиск данного ключа в массиве, на который указывает `base` (который состоит из `num` элементов, каждый из которых имеет размер в байтах),

и возвращает указатель `void *` на соответствующий элемент, если он найден.

Для выполнения поиска функция выполняет серию вызовов для сравнения с ключом в качестве первого аргумента и элементами массива, на которые указывает `base` в качестве второго аргумента.

### **Выполнение работы.**

Считываем входные данные в массив *text* и *str* с помощью функции `fgets`. С помощью функции `strtok` разделяем текст на слова (разделители – пробелы и точки), записываем их в массив *words*.

Сортируем слова в *words*, используя алгоритм быстрой сортировки (*qsort*) для дальнейшего поиска *str* в тексте. С помощью функции `bsearch` определяем, есть ли во входных данных слово *str*, если таковое существует – выводим “exists”, в противном случае выводим “doesn’t exist”.

Освобождаем память 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.	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. is	exists	программа работает корректно
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. ally	doesn't exist	программа работает корректно
3.	Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems Java platform. originally	exists	программа работает корректно

## **Выводы.**

Были изучены функции стандартной библиотеки Си, написана программа сортировки текста и поиска в нем определенного слова.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: *main.c*

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

int compare(const void *a, const void *b){
    return strcmp(*(char**)a, *(char**)b);
}

int main(){
    char *text = malloc(1001 * sizeof(char));
    char **words = malloc(1001 * sizeof(char*));
    char *str = malloc(31 * sizeof(char));
    int words_in_text = 0;
    fgets(text, 1001, stdin);
    fgets(str, 31, stdin);
    text[strlen(text)] = '\0';
    str[strlen(str)] = '\0';
    char *word = strtok(text, " .");
    while(word != NULL){
        words[words_in_text] = word;
        words_in_text++;
        word = strtok(NULL, " .");
    }
    qsort(words, words_in_text, sizeof(char**), compare);
    char **arg = (char**) bsearch(&str, words, words_in_text,
sizeof(char**), compare);
    if(arg != NULL)
        printf("exists");
    else
        printf("doesn't exist");
    free(words);
    free(text);
    free(str);
    return 0;
}
```