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. 1304 _____ Ярусова Т. В.

Преподаватель _____ Чайка К. В.

Санкт-Петербург

2022

Цель работы.

Освоить работу со стандартными библиотека языка С и научиться использовать функции данных библиотек в своих программах.

Задание.

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
- определить, присутствует ли в тексте str, используя алгоритм двоичного поиска
- вывести строку "exists", если str в тексте есть и "doesn't exist" в противном случае.

Основные теоретические положения.

Для реализации алгоритма быстрой сортировки и алгоритма двоичного поиска необходимо использовать функции стандартной библиотеки.

Выполнение работы.

Для выполнения поставленной задачи, необходимо подключить дополнительные библиотеки. Библиотеку <*stdlib.h*> и <*string.h*>.

В главной функции *int main()* объявляется массив символов *char text* размером 1000, в котором будет храниться вводимый пользователем текст, и массив символов *char str* размером 30, в котором будет храниться вводимая пользователем строка, которая будет искаться в вводимом пользователем тексте. С помощью функции *fgets()* считываются вводимый пользователем текст в *text* и вводимая пользователем строка в *str*.

В *char* str_new* сохраняется строка *str* без пробелов и знаков переноса строки с помощью функции *strtok()*.

С помощью цикла *while()* и функции *strtok()* массив *text* разбивается на строки, не содержащие точку и пробелы. Данное разбиение сохраняется в массив строк *words*.

С помощью функции *qsort()* происходит сортировка массива words. В функцию подается первым аргументом массив, который необходимо отсортировать, вторым аргументом – длина данного массива, третьим аргументом – размер одного элемента массива и последним аргументом – функция *cmp*.

С помощью функции *bsearch()* происходит бинарный поиск элемента *str_new* в массиве *words*. В функцию подается первым аргументом элемент, который необходимо найти в массиве, вторым аргументом – массив, в котором необходимо найти данный элемент, третьим аргументом – длина данного массива, четвертым аргументом – размер одного элемента массива и последним аргументом – функция *cmp*.

Результат работы функции *bsearch()* сохраняется в переменную *char** *result*.

Если из функции *bsearch()* вернулся указатель в переменную *result*, то с помощью функции *puts()* выводится строка *“exists”*, иначе выводится *“doesn’t exist”*.

Функция *return 0* заканчивает работу программы.

Функции:

В функцию *int cmp(const void* a, const void* b)* подаются на вход в качестве аргументов указатели на переменные типа const void. В функции происходит разыменование переменных *a* и *b* и присваивание им типа *const char***, т.к передаваемые значения – указатели на массив массивов строк. С помощь функции *strcmp()* сравниваются две строки и из функции *cmp*, с помощью функции *return* возвращается значение, которое возвращается из функции *strcmp()*.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1	I am student LETI. am	exists
2	I am student LETI. leti	“doesn’t exist”

Выводы.

В ходе выполнения лабораторной работы были использованы функции стандартных библиотек языка программирования С.

Разработана программа, выполняющая считывание с клавиатуры текста и строки, которую необходимо найти в данном тексте. Исходный текст был разбит на слова, отсортирован с помощью функции *qsort()* и введенная пользователем строка была найдена или не найдена в тексте с помощью функции бинарного поиска *bsearch()*. Результат программы был выведен на экран.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Yarusova_Tatyana_lb1

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

int cmp(const void* a, const void* b){
    const char** word_1 = (const char**) a;
    const char** word_2 = (const char**) b;
    return strcmp(*word_1, *word_2);
}

int main(){
    char text[1000];
    char str[30];
    fgets(text, sizeof(char)*1000, stdin);
    fgets(str, sizeof(char)*30, stdin);
    char* str_new = strtok(str, " .\n");


    char* word = strtok(text, " .");
    int count_word = 0;
    char** words;
    words = malloc(sizeof(char*) * count_word);
    while(word){
        count_word++;
        words = realloc(words, sizeof(char*) * count_word);
        words[count_word-1] = word;
        word = strtok(NULL, " .");

    }

    qsort(words, count_word, sizeof(char*), cmp);

    char**    result    =    bsearch(&str_new,    words,    count_word,
sizeof(char*), cmp);

    free(words);

    if(result)
        puts("exists");
    else
        puts("doesn't exist");

    return 0;
}
```