**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: **Стандартная библиотека языка С.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Рыбин А.С. |
| Преподаватель |  | Кринкин К.В. |

Санкт-Петербург

2017

Оглавление

[**Оглавление** 2](#_Toc476218818)

[Цель: 3](#_Toc476218819)

[Задание: 3](#_Toc476218820)

[Содержание: 3](#_Toc476218821)

[Разбиение исходного текста на слова 3](#_Toc476218822)

[Сортировка последовательности слов 3](#_Toc476218823)

[Бинарный поиск 3](#_Toc476218824)

[Вывод: 3](#_Toc476218825)

[ПРИЛОЖЕНИЕ 1 4](#_Toc476218826)

Цель:

Написание программы с использованием функций стандартной библиотеки языка Си

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 (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.

Содержание:

Разбиение исходного текста на слова

/\* Вспомогательный указатель для разбиения на слова \*/

char\* pointer = strtok(buffer, " .");

while (pointer)

{

strcpy(words\_array[count], pointer);

pointer = strtok(NULL, " .");

count++;

}

Сортировка последовательности слов

/\* Сортировка массива \*/

qsort(words\_array, count, 1000, compare);

Бинарный поиск

/\* Бинарный поиск \*/

char\* ptr = (char\*)bsearch(substring, words\_array, count, 1000, compare);

if (ptr == NULL)

printf("doesn't exist\n");

else

printf("exists\n");

**Вывод:**

Данная лабораторная работа была выполнена исключительно с помощью функций стандартной библиотеки (string.h, stdlib.h), благодаря этому отсутствовала необходимость реализации собственных процедур; заметно уменьшился объем кода и увеличилось быстродействие программы за счет использования проверенных готовых решений.

ПРИЛОЖЕНИЕ 1

/\*

Лабораторная работа №1. Второй семестр. Рыбин Александр гр.6304

\*/

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

/\* Функция сравнения для сортировки и бинарного поиска \*/

int compare(const void\* a, const void\* b)

{

return strcmp((char\*)a, (char\*)b);

}

int main() {

/\* Буфер для хранения исходного текста \*/

char buffer[1000];

/\* Искомое слово \*/

char substring[30];

/\* Массив хранения слов \*/

char words\_array[1000][1000];

int count = 0;

fgets(buffer, 1000, stdin);

(\*strstr(buffer, "\n")) = '\0';

fgets(substring, 30, stdin);

/\* Вспомогательный указатель для разбиения на слова \*/

char\* pointer = strtok(buffer, " .");

while (pointer)

{

strcpy(words\_array[count], pointer);

pointer = strtok(NULL, " .");

count++;

}

/\* Сортировка массива \*/

qsort(words\_array, count, 1000, compare);

/\* Бинарный поиск \*/

char\* ptr = (char\*)bsearch(substring, words\_array, count, 1000, compare);

if (ptr == NULL)

printf("doesn't exist\n");

else

printf("exists\n");

system("pause");

return 0;

}