**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 |  | Виноградов К.А. |
| Преподаватель |  | Кринкин К.В. |

Санкт-Петербург

2017

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.

Ход работы:

* Разбиение исходного текста на слова

char\* token=strtok(text, " ."); //разделение текста на слова

while(token!=NULL)

{

int numb=strlen(token); //подсчет длины каждого слова

strncpy(words[i], token, numb); //копирование в массив слов

token=strtok(NULL, " ,");

i++;

}

* Сортировка массива слов

qsort(words, i, 1000, cmp); //функция быстрой сортировки

* Бинарный поиск в сортированном массиве

char\* pointer=(char\*)bsearch(str, words, i, 1000, cmp); //функция //бинарного

//поиска

Вывод:

Данная лабораторная работа была выполнена исключительно с помощью функций стандартной библиотеки (string.h, stdlib.h), благодаря этому нам не потребовалось реализовывать собственные функции, вследствие чего заметно уменьшился объем работы.

ПРИЛОЖЕНИЕ 1

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int cmp(const void\* p1, const void\* p2) //реализация функции сравнения для

{ //строк

return strcmp((char\*)p1, (char\*)p2);

}

int main()

{

int counter=0;

int i=0; //счетчики и переменные

int max=0;

char text[1000]; //массив для текста на ввод

char str[30]; //массив для искомого слова

char words[500][1000]; //массив для разбиения текста на слова

fgets(text, 1000, stdin); //счет с входного потока в первые

(\*strstr(text, "\n"))='\0'; //два массива

fgets(str, 30, stdin);

char\* token=strtok(text, " ."); //разделение текста на слова

while(token!=NULL)

{

int numb=strlen(token); //подсчет длины каждого слова

strncpy(words[i], token, numb); //копирование в массив слов

token=strtok(NULL, " ,");

i++;

}

qsort(words, i, 1000, cmp); //функция быстрой сортировки

char\* pointer=(char\*)bsearch(str, words, i, 1000, cmp); //функция бинарного

//поиска

if(pointer!=NULL)

{

printf("exists\n");

}

else //вывод результата поиска

{

printf("doesn't exist");

}

return 0;

}