МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное

образовательное учреждение

высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

**по дисциплине**

**“ Алгоритмы построение и анализ”**

Выполнил: студент гр. ФИб-3302-51-00 ё. \_\_\_\_\_\_\_\_\_\_\_

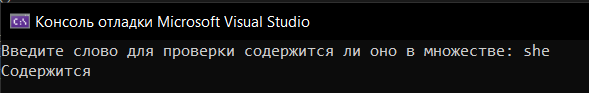
Проверил: доцент кафедры ПМиИ Разова Е. В. \_\_\_\_\_\_\_\_\_\_\_

Киров 2020

***Задание 1. Реализовать построение бора по заданному множеству слов М.***

***Задание 2. Поиск в боре.***

Дано множество слов М и слово S. Используя поиск в боре, определить, содержится ли слово S во множестве М.



int main()

{

setlocale(0, "");

string keys[] = { "he", "she", "his", "her"};

//string keys[] = { "oh", "oha", "okho","hoga", "hora", "horka"};

int n = sizeof(keys) / sizeof(keys[0]);

struct TrieNode \*root = getNode();

// задание 1: посторить Бор

for (int i = 0; i < n; i++)

insert(root, keys[i]);

// задание 2: поиск слова

string pattern;

cout << "Введите слово для проверки содержится ли оно в множестве: ";

cin >> pattern;

search(root, pattern) ? cout << "Содержится\n" :

cout << "Не содержится\n";

return 0;

}

const int ALPHABET\_SIZE = 26;

// бор структура

struct TrieNode

{

struct TrieNode \*children[ALPHABET\_SIZE];

// isEndOfWord истинно, если узел представляет конец слова

bool isEndOfWord;

};

// Возвращает новый узел Бора

struct TrieNode \*getNode(void)

{

struct TrieNode \*pNode = new TrieNode;

pNode->isEndOfWord = false;

for (int i = 0; i < ALPHABET\_SIZE; i++)

pNode->children[i] = NULL;

return pNode;

}

// Если ключ является префиксом узла trie, отмечаем листовой узел

// Если нет, вставляет ключ в дерево

void insert(struct TrieNode \*root, string key)

{

//Создаем дерево из одной вершины (в нашем случае корня), затем добавляем элементы в дерево.

//Добавляем слова одно за другим. Следуем из корня по рёбрам, отмеченным буквами из Pi

//Если заканчивается в v, отмечаем вершину v как лист.

struct TrieNode \*pCrawl = root;

for (int i = 0; i < key.length(); i++)

{

int index = key[i] - 'a';

if (!pCrawl->children[index])

pCrawl->children[index] = getNode();

pCrawl = pCrawl->children[index];

}

//пометить последний узел как лист

pCrawl->isEndOfWord = true;

}

// Поиск слова

bool search(struct TrieNode \*root, string key)

{

struct TrieNode \*pCrawl = root;

for (int i = 0; i < key.length(); i++)

{

int index = key[i] - 'a';

if (!pCrawl->children[index])

return false;

pCrawl = pCrawl->children[index];

}

return (pCrawl != NULL && pCrawl->isEndOfWord);

}