

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**

Институт искусственного интеллекта

Кафедра математического обеспечения и стандартизации

информационных технологий

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3**

**Стандартные типы данных языка программирования С++ для представления текстовых**

**данных**«СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ»

|  |  |
| --- | --- |
| Выполнил студень группы ИНБО-02-21      Принял старший преподаватель | Юдов С.А. |
| кафедры МОСИТ | Скворцова Л.А. |
| Практическая работа выполнена «\_\_»\_\_\_\_\_\_\_2022г. | \_\_\_\_\_\_\_\_\_ |
|  |  |
| «Зачтено» «\_\_»\_\_\_\_\_\_\_2022г. | \_\_\_\_\_\_\_\_\_ |
|  |  |

Москва 2022

**Содержание**

[Задание 1 3](#_Toc99285529)

[1. Условие задачи и варианта 3](#_Toc99285530)

[2. Разработка задачи 3](#_Toc99285531)

[3. Определение функций 4](#_Toc99285532)

[4. Реализация функций 4](#_Toc99285533)

[5. Кодирование алгоритма программы 5](#_Toc99285534)

[6. Таблица тестов программы 6](#_Toc99285535)

[Задание 2 7](#_Toc99285536)

[1. Определение функций 7](#_Toc99285537)

[2. Декомпозиция 7](#_Toc99285538)

[3. Реализация функций 7](#_Toc99285539)

[4. Кодирование алгоритма программы 8](#_Toc99285540)

[5. Таблица тестов программы 9](#_Toc99285541)

[Вывод 10](#_Toc99285542)

[Список информационных источников 11](#_Toc99285543)

# Задание 1

## 1. Условие задачи и варианта

* 1. Дан текст, содержащий слова, которые разделены пробелами. Найти слова в этом предложении, которые состоят из тех же букв, что и другое слово предложения. Например, ток, кот; лес, сел; гора, рога.

## 2. Разработка задачи

2.1. Постановка задачи.

2.1.1. Дано. Строка со словами разделённые пробелом

2.1.2. Результат. Слова-анаграммы выписанные через пробел

2.1.3. Ограничения. Реализация определённым способом

2.2. Описание модели решения

Модель решения заключается в разделении слов и занесения их в массив, для удобного перебора с помощью функции. Функция принимает массив и начинает перебор. Если отсортированная строка, которая находится в массиве является такой же, как и другая. Слово считается верным и заносится в вектор или выводится в консоль. На выходе получаются только слова-анаграммы.

2.3. Декомпозиция – список алгоритмов, которые требуются разработать в соответствии исследованной моделью

2.3.1. Список подзадач:

1. Создание массива слов из строки
2. Проверка каждого слова с помощью цикла с сортировкой и вывод их в консоль

2.3.2. Определение прототипов функций:

1. Создание массива слов из строки

Предусловие. char \*\*arr – указатель на статичный массив, char \*str – указатель на строку, char \*s – указатель на пробельный символ

Постусловие. Заполненный массив из k элементов

int arrсreate(char \*\*arr, char \*str, char \*s)

1. Поиск и вывод слов-анаграмм

Предусловие char \*\*arr – указатель на статичный массив, int k > 0 – количество

Постусловие. Вывод слов-анаграмм

void checkingallarray(char \*\*arr, int k)

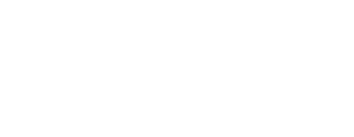
## 3. Определение функций

3.1. Создание массива слов из строки:

int



A



char \*\*arr

char \*str

char \*s



Входные данные



Результат

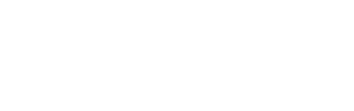


int arrсreate

3.2. Поиск и вывод слов-анаграмм



A



char \*\*arr

int k



Входные данные



Результат



void checkingallarray

## 4. Реализация функций

int arrсreate(char \*\*arr, char \*str, char \*s) {

int i = 0;

while (arr[i] != NULL) // проход по строке

{

// puts(arr[i]); // вывод слова

arr[++i] = strtok(NULL, s); // запись слова в двумерный массив

}

return i; // возвращение количества переменных в массиве

}

void checkingallarray(char \*\*arr, int k) {

for (int i = 0; i < k; i++) { // проход по элементам массива

for (int j = 0; j < k; j++) {

char temp1[strlen(arr[i])];

strcpy(temp1, arr[i]);

char temp2[strlen(arr[j])];

strcpy(temp2, arr[j]);

sort(temp1, temp1 + strlen(temp1) \* sizeof(char));

sort(temp2, temp2 + strlen(temp2) \* sizeof(char));

if (strcmp(temp1, temp2) == 0 && i != j) {

printf("% s ", arr[i]);

}

}

}

}

## 5. Кодирование алгоритма программы

#include <iostream>

#include <cstdio>

#include <string>

#include <cstring>

#include <algorithm>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");;

char str[100] = "";

char \*arr[100];

char s[] = " ";

int k = 0;

cin.clear();

gets(str);

arr[0] = strtok(str, s); // инициализация массива слов с разделителем - пробелом

k = arrсreate(arr, str, s);

checkingallarray(arr, k);

return 0;

}

## 6. Таблица тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер задачи | Исходные данные | Ожидаемый результат | Результат программы | Тест пройден/не пройден |
| 1 | cat tac ssa ddd | cat tac | cat tac | Тест пройден |
| 2 | man www nam | man nam | man nam | Тест пройден |

# Задание 2

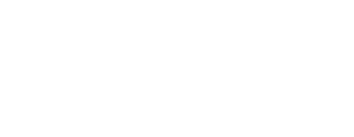
## 1. Определение функций

1.1. Создание массива слов из строки:

vector<string>



A



string str



Входные данные



Результат



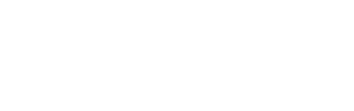
vector<string> arrсreate

1.2. Поиск слов-анаграмм

vector<string>



A



vector<string> a



Входные данные



Результат

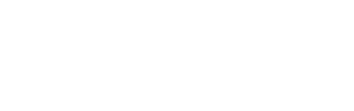


vector<string> checkingallarray

1.3. Вывод вектора со словами-анаграммами



A



vector<string> &x



Входные данные



Результат



void outputvector

## 2. Декомпозиция

* 1. Задачу следует разбить на следующие подзадачи:

1. Создание вектора слов из строки
2. Проверка каждого слова с помощью цикла с сортировкой
3. Вывод слов-анаграмм в консоль

## 3. Реализация функций

vector<string> arrсreate(string str) {

vector<string> temp;

string s = "";

int k = 0;

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

if (str[i] != ' ') {

s = s + str[i];

} else {

if (s != "") {

temp.push\_back(s);

s = "";

}

} }

temp.push\_back(s);

return temp;

}

void outputvector(vector<string> &x) {

for (auto const &element: x) {

cout << element << ' ';

}

cout << endl;

}

vector<string> checkingallarray(vector<string> a) {

vector<string> temp;

for (int i = 0; i < a.size(); i++) { // проход по элементам массива

for (int j = 0; j < a.size(); j++) {

string temp1 = a[i];

string temp2 = a[j];

sort(temp1.begin(), temp1.end(), greater<>());

sort(temp2.begin(), temp2.end(), greater<>());

if (temp1 == temp2 && i != j) {

temp.push\_back(a[i]);

}

}

}

return temp;

}

## 4. Кодирование алгоритма программы

#include <iostream>

#include <string>

#include <algorithm>

#include <vector>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");;

string str;

getline(cin, str);

vector<string> z = arrсreate(str);

vector<string> result = checkingallarray(z);

outputvector(result);

return 0;

## 5. Таблица тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Исходные данные | Ожидаемый результат | Результат программы | Тест пройден/не пройден |
| 1 | cat tac ssa ddd | cat tac | cat tac | Тест пройден |
| 2 | man www nam | man nam | man nam | Тест пройден |

# Вывод

При выполнении практической работы были освоены навыки обработки null терминальных строк и строк класса String. Были разработаны программные решения задач, предложенных в соответствующем варианте.

# Список информационных источников

* Учебник по C++ [http://www.lmpt.univ-tours.fr/~volkov/C++.pdf](https://d.docs.live.net/793bf2b94bcfd004/Рабочий%20стол/http:/www.lmpt.univ-tours.fr/~volkov/C++.pdf)
* Документация по языку C++ [https://docs.microsoft.com/ruru/cpp/?view=msvc-160](https://d.docs.live.net/793bf2b94bcfd004/Рабочий%20стол/https:/docs.microsoft.com/ruru/cpp/?view=msvc-160)