**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**Отчет По Дисциплине**

**«Алгоритмизация и программирование»**

**Лабораторная работа №3**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-115

Клепиков Степан Даниилович

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ФИО)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ФИО)

Москва 2022 г.

**Задание**

Задан текст. Напечатать все слова, отличные от последнего слова, предварительно преобразовав каждое из них по следующему правилу: перенести последнюю букву в начало слова. Использовать функции.

1. Таблица имён:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Рабочие переменные | | | | | |
| i | integer | | | Счетчик | |
| j | integer | | | Счетчик | |
| numb | integer | | | Переменная выбора варианта | |
| s | integer | | | Символ файла | |
| numb | vector | | | Временная переменная | |
| name | string | | | Имя создаваемого файла | |
| size | integer | | | Кол-во вводимых символов | |
| num | integer | | | Вводимый символ | |
| Исходные данные | | | | | |
| arr | | integer | | | Исходный файл |
| Результат | | | | | |
| arr2 | | | integer | | Файл с четными числами |
| arr3 | | | integer | | Файл с нечетными числами |

1. Блок-схема:

(внешние функции)

1. Отладочный пример:

Я не хочу умирать

1. Код программы:

#include <iostream>//подключение библиотек

#include <algorithm>

#include <vector>

#include <string>

using namespace std;

string last\_word(const string& str) //функция поиска последнего слова

{

if (str.empty())//проверка пустую строчку

{

cout << "No string\n";//вывод

}

int i = str.length() - 1;

while (i >= 0 && str[i] != ' ')

{

i--;

}

string last\_word;

for (int j = i + 1; j < str.length(); j++)

{

last\_word += str[j];

}

return last\_word; // возврат функцией последнего слова

}

void SplitString(string s, vector<string>& v) {

string temp = "";

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

if (s[i] == ' ') {

v.push\_back(temp);

temp = "";

}

else {

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

}

}

v.push\_back(temp);

}

void swapchar(string& a) {

a.insert(0, 1, a.back());

a.pop\_back();

}

void rev\_swapchar(string& a) {

a = a + a[0];

a.erase(0, 1);

}

int main() {

cout << "Enter text : \n";

string str, lst;

vector<string> v;

getline(cin, str);

lst = last\_word(str);

SplitString(str, v);

v.pop\_back();

for (auto i : v) {

swapchar(i);

if (i != lst) {

rev\_swapchar(i);

cout << "result is : " << i << "\n";

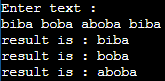
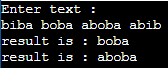
}

}

return 0;

}

1. Результат выполнения работы программы:



1. Вывод:

В ходе выполнения работы были изучены базовые алгоритмы и функции языка высокого уровня C++.

Был произведен анализ их работы, на основании которого была составлена блок-схема. На контрольных примерах было выяснено, что все написанные функции работают корректно и в коде отсутствуют ошибки.

Был оформлен комплект документации на программный код.