**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление информатика и вычислительная техника

Отчет

по лабораторной работе №3

по дисциплине

**«**ПРОГРАММИРОВАНИЕ**»**

Выполнил:

Студент группы 8В31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Р. Колесников

Проверил:

Программист \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Д. Воеводин

**Цель работы:**получить навыки работы с базовым контейнером string на языке программирования С++.

**Задание:** Реализовать функции к заданиям, представленным ниже. Собрать и отладить проект на языке программирования С++ с применением CMake в VS Code по реализации длинной арифметики на основе базового контейнера vector. Ввести с клавиатуры два положительных целых числа: input\_1 и input\_2, при этом они больше 64. Вычислить 2 input\_1 + 2 input\_2. Ответ напечатать в терминале.

**Вывод:** получены навыки работы с базовым контейнером string на языке программирования С++.

**Приложение 1**

#include <iostream>

#include <string>

#include <vector>

/\*\*

 \* @brief ФУНКЦИЯ ДЛЯ ПЕРВОЙ ЗАДАЧИ

 \* функция изменяет символы "." на комбинацию символов "[.]"

 \*

 \* @param S - входная строка содержащая IP

 \* @return std::string - входная строка с изменёнными символами "." на "[.]"

 \*/

std::string editIP(std::string S)

{

    std::string result = "";

    for(int i = 0; i < S.size(); i++)

    {

        if(S[i] == '.')

        {

            S.erase(i, 1);

            S.insert(i, "[.]");

            i += 1;

        }

    }

    return S;

}

/\*\*

 \* @brief ФУНКЦИЯ ДЛЯ ВТОРОГО ЗАДАНИЯ

 \* функция проверяет строку на палиндром(то есть что строка читается одинаково как в одну, так и в другую сторону)

 \*

 \* @param S - исходная строка

 \* @return true - если строка палиндром

 \* @return false - если строка не является палиндромом

 \*/

bool isPalindrome(std::string S)

{

    //УДАЛЯЕМ ПРОБЕЛЫ

    for(int i = 0; i < S.size(); i++)

    {

        if(S[i] == ' ')

        {

            S.erase(i, 1);

        }

    }

    bool flag = 1;

    for(int i = 0; i < S.size() / 2; i++)

    {

        if(S[i] != S[S.size() - 1 - i])

        {

            flag = 0;

        }

    }

    return flag;

}

/\*\*

 \* @brief функция возвращает максимальный счёт среди множества слов составленных из вектора letters

 \*

 \* @param words - вектор строк (вектор слов которые нужно будет составлять из вектора letters)

 \* @param letters - вектор символов, они могут повторяться

 \* @param score - количество очков за каждую букву латинского алфавита

 \* @return int - максимальное количество очков со всех слов вектора words

 \*/

int checkScore(std::vector<std::string> words, std::vector <char> letters, std::vector<int> score){

    int char2[26] = {0};

    int maxi = 0;

    for (int i = 0; i < (int)letters.size(); ++i)

    {

        char2[letters[i] - 'a']++;

    }

    for (size\_t i = 0; i < words.size(); ++i)

    {

        bool fl = true;

        std::string word = words[i];

        int str2[26] = {0};

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

        {

            str2[word[j] - 'a']++;

        }

        for (int k = 0; k < 26; ++k)

        {

            if (char2[k] < str2[k])

            {

                fl = false;

                break;

            }

        }

        if (!fl)

        {

            continue;

        }

        int a = 0;

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

        {

            a += score[word[j] - 'a'];

        }

        if (maxi < a)

        maxi = a;

    }

    return maxi;

}

int main(){

    //ПЕРВАЯ ЗАДАЧА

    std::string test\_one = "255.255.255.254";

    std::string test\_two = "203.0.113.0";

    std::string result = editIP(test\_one);

    std::cout << result;

    //ВТОРАЯ ЗАДАЧА

    std::string S;

    std::cout << "Введите строку: ";

    getline(std::cin, S);

    bool res = isPalindrome(S);

    std::cout << res;

    //ТРЕТЬЯ ЗАДАЧА

    std::vector<std::string> words = {"cat", "dog", "pen", "fog"};

    std::vector<char> letters = {'a', 'c', 't', 'o', 'g', 'p', 'e', 'n', 'f'};

    std::vector<int> score = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};

    std::cout << "Самое большое количество очков: " << checkScore(words, letters, score);

    return 0;

}

Ссылка на Github: https://github.com/tu9ric/Lab\_3\_Programming