Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

# ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10

з навчальної дисципліни “Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ ВВЕДЕННЯМ/ВИВЕДЕННЯМ

ВИКОНАВ

студент академічної групи

КІ-21-1 Назаревський М. В.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Усік П. С.

Кропивницький – 2022

# ЛАБОРАТОРНА РОБОТА №10

**Тема:** Реалізація програмних модулів, оброблення даних складових типів з файловим введенням/виведенням.

**Мета роботи:** полягає у набутті ґрунтовних вмінь і практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування С++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

## Варіант 1

**Завдання:**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як

складові статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище лабораторних робіт №8–9).

2. Реалізувати тестовий драйвер автоматизованої перевірки рограмних модулів розв’язування задач 10.1–10.3.

**ХІД РОБОТИ**

**Завдання 1**

## 

Рисунок 1.1 - Умова задачі 10.1

### **Аналіз задачі**

Вхідні дані: Довільний текст українською мовою.

Вихідні дані: Авторська інформація, кількість символів у вхідному файлі, повідомлення про наявність слів “програма”, “модуль”, “студент”, “програміст”.

### **Проектування програмного модуля:**

За даними задачі, користувач вводить довільний текст у вхідний текстовий файл – inputFile.txt.

Потрібно додати авторську інформацію у вихідний файл – outputFile.txt.

Потрібно реалізувати систему підрахунку символів у вхідному файлі, та відобразити це у вихідному файлі – outputFile.txt.

Також потрібно потрібно додати повідомлення у файл outputFile.txt, про наявність слів “програма”, “модуль”, “студент”, “програміст” та виконати розрізнення від інших слів які мають їх у собі, наприклад “студент” - “студентка”.

**Завдання 2**

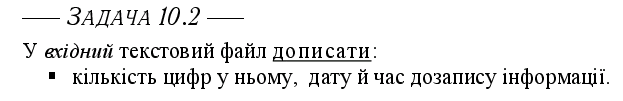


Рисунок 1.2 - Умова задачі 10.2

### **Аналіз задачі:**

Вхідні дані: Довільний текст українською мовою.

Вихідні дані: Кількість цифр у файлі та дата й час дозапису інформації до нього.

### **Процедурна алгоритмізація:**

За даними задачі, користувач вводить довільний текст у вхідний текстовий файл – inputFile.txt.

Розробляємо масив цифр від 0 до 9, а потім порівнюємо кожен елемент масиву з текстом який ввів користувач, якщо рівність вірна, то створена раніше змінна, яка слугує лічильником – збільшується на 1.

Додаємо до вхідного файлу значення лічильника.

З допомогою бібліотеки ctime, додаємо до вхідного текстового файлу дату та час дозапису інформації до нього.

**Завдання 3**

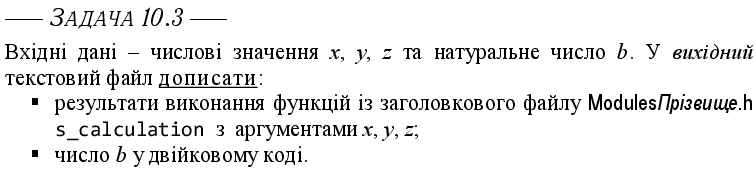


Рисунок 1.3 - Умова задачі 10.3

### **Аналіз задачі:**

Вхідні дані: x, y, z - дійсні знакові числа, натуральне число b.

Вихідні дані: Результат виконання функції s\_calculation, яка була створена у результаті виконання лабораторної роботи №8. Число b у двійковому коді (можна скористатись функцією із лабораторної роботи №9).

### **Процедурна алгоритмізація:**

За даними задачі, користувач вводить числові значення x, y, z та натуральне число b у вхідний текстовий файл – inputFile.txt.

Додаємо результат виконання функції s\_calculation у вихідний текстовий файл – outputFile.txt.

Використовуємо функцію з минулої лабораторної роботи, яка перетворює число у двійковий код та додаємо у вихідний текстовий файл – outputFile.txt.

**Результати виконання тестового модуля**

Тест завдання 10.1 Тест завдання 10.2 Тест завдання 10.3

Passed Passed Passed

Passed Passed Passed

Passed Passed Passed

Passed Passed Passed

Passed Passed Passed

# ВИСНОВКИ

Під час виконання лабораторної роботи № 10 на тему **«**Реалізація програмних модулів, оброблення даних складових типів з файловим введенням/виведенням**»** з предмету «Базові Методології та Технології Програмування» я відпрацював на практиці створення статичних бібліотек та власних модулів і заголовних файлів, написання власних текстових драйверів для своїх модулів. Набув ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

У ході виконання даної лабораторної роботи були реалізовані функції розв’язування задач 10.1 – 10.3 як складових статичної бібліотеки libModulesNazarevskyi.a (проект ModulesNazarevskyi, створений під час виконання лабораторної роботи №8 та доповнений під час виконання лабораторної роботи №9) та був реалізований тестовий драйвер для автоматизованої перевірки програмних модулів розв'язування задач 10.1 – 10.3 зі статичної бібліотеки libModulesNazarevskyi.a

Аналізувавши завдання 1, я чітко для себе зазначив, що потрібно реалізувати підрахунок символів без пропусків (пробілів) та систему пошуку слів, яка зможе чітко відлічити слово від слів, що містять у собі це слово.

Успішно виконавши завдання 1, я завантажив папку ModulesNazarevskyi до свого репозитарію на GitHub.

Завдання 2 передбачало створення функції, яка підраховує числа у вхідному файлі, а також виводить дату й час дозапису інформації до нього.

Створивши масив із цифр від 0 до 9, я створив вкладений цикл та умову, що порівнює вхідний текст з масивом цифр, у результаті рівності – лічильник збільшувався на 1, тим самим він відображав кількість цифр у файлі.

Вивести дату й час дозапису інформації у вхідному файлі, мені вдалось за допомогою бібліотеки ctime.

Успішно виконавши завдання 2, я завантажив оновлену папку ModulesNazarevskyi до свого репозитарію на GitHub.

Завдання 3 ставило за мету відображення результату функції s\_calculation у вихідному текстовому файлі, значення для функції беруться з вхідного файлу. Також реалізувати функцію котра перетворює число b, задане у вхідному файлі, на двійковий код.

Використавши функцію s\_calculation, яку я створив під час виконання лабораторної роботи №8, я адаптував її для роботи у текстовому файлі.

Скориставшись бібліотекою bitset, яку я відкрив для себе під час виконання лабораторної роботи №9, я успішно виконав адаптацію для роботи з текстовими файлами.

Успішно виконавши завдання 3, я завантажив оновлену папку ModulesNazarevskyi до свого репозитарію на GitHub.

Я розробив TestDriver, котрий мав автоматично перевірити працездатність створених мною модулів, заздалегідь створивши unit test'и, я записав до тест-драйверу значення з unit test'ів . У результаті виконання, помилок не виявлено, тобто модулі справні.

Також мені довелось змінити формат текстових файлів з ANSI на UTF-8, задля коректного відображення на сайті Github.

Лабораторна робота №10 була успішно виконана мною та завантажена до власного репозитарію.

**ДОДАТОК А**

**Лістинг ModulesNazarevskyi**

#include <iostream>

#include <cmath>

#include <bitset>

#include <string>

#include <sstream>

#include <fstream>

#include <cstring>

#include <ctime>

#include "ModulesNazarevskyi.h"

;

using namespace std;

double s\_calculation(float x, float y, float z)

{

return z\*sin(pow(x,2) \* y) + sqrt(abs(z - x\*12))/pow(y,3);

}

double bofort(int bal)

{

switch (bal)

{

case 0:

cout << "< 0.3 - Штиль";

return 0.3;

break;

case 1:

cout << "0.3 - 1.5 - Тихий";

return 1.5;

break;

case 2:

cout << "1.5 - 3.3 - Легкий";

return 3.3;

break;

case 3:

cout << "3.3 - 5.4 - Слабкий";

return 5.4;

break;

case 4:

cout << "5.4 - 7.9 - Помірний";

return 7.9;

break;

case 5:

cout << "7.9 - 10.7 - Свіжий";

return 10.7;

break;

case 6:

cout << "10.7 - 13.8 - Сильний";

return 13.8;

break;

case 7:

cout << "13.8 - 17.1 - Міцний";

return 17.1;

break;

case 8:

cout << "17.1 - 20.7 - Дуже міцний";

return 20.7;

break;

case 9:

cout << "20.7 - 24.4 - Шторм";

return 24.4;

break;

case 10:

cout << "24.4 - 28.4 - Сильний шторм";

return 28.4;

break;

case 11:

cout << "28.4 - 32.6 - Шквальний шторм";

return 32.6;

break;

case 12:

cout << ">32.6 - Ураган(Буревій)";

return 32.7;

break;

default:

cout << "Невірний бал Бофорта, введіть число від 0 до 12";

return -1;

}

}

powers powAndNum(int n, int m)

{

powers p;

p.n = pow(n, m);

p.m = pow(m, n);

cout << "Непарні числа у проміжку: ";

for (int i = n; i <= m; ++i){

if (i%2){

cout << i <<" ";

}

}

cout << endl;

cout << "Число n в степені m = " << p.n << endl;

cout << "Число m в степені n = " << p.m << endl;

return p;

}

int numBit(int num)

{

if(num < 0 || num > 80000){

cout << "Невірне число, введіть число від 0 до 80000 " << endl;

return -1;

}

bitset<32> b\_number(num);

if(b\_number[2]){

return b\_number.count();

}

else {

return 32 - b\_number.count();

}

}

//10

void canToRedaction(string outputFileName, string inputFileName)

{

ofstream inputFile(inputFileName);

ofstream outputFile(outputFileName);

if (!inputFile || !outputFile) {

cout << "Помилка, неможливо відкрити файл для редагування\a" << endl;

inputFile.close();

outputFile.close();

exit(1);

}

inputFile.close();

outputFile.close();

}

bool inputTxtFile(string inputFileName, string inputText)

{

ofstream inputFile(inputFileName, ios::out);

inputFile << inputText << endl << endl;

inputFile.close();

return true;

}

// Task 10.1

bool myInfo(string outputFileName)

{

ofstream outputFile(outputFileName, ios::out);

outputFile << "Виконавець: Назаревський Максим Володимирович" << endl

<< "Рік розробки: 2022" << endl

<< "Країна, місто: Україна, Кропивницький" << endl

<< "ВНЗ: Центральний Національний Технічний Університет" << endl << endl;

outputFile.close();

return true;

}

int numberSymbolsInText(string outputFileName, string inputTextInFile)

{

ofstream outputFile(outputFileName, ios::app);

int numberSymbols = 0;

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

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

numberSymbols++;

}

}

outputFile << "Кількість символів з вхідного файлу: "

<< numberSymbols << endl << endl;

outputFile.close();

return numberSymbols;

}

bool wordsInText(string outputFileName, string inputTextInFile)

{

ofstream outputFile(outputFileName, ios::app);

size\_t word1 = inputTextInFile.find("програма ");

size\_t word2 = inputTextInFile.find("модуль ");

size\_t word3 = inputTextInFile.find("студент ");

size\_t word4 = inputTextInFile.find("програміст ");

if (word1 != string::npos)

outputFile << "Знайдено слово 'програма'" << endl << endl;

else

outputFile << "Cлово 'програма' - не знайдено" << endl << endl;

if (word2 != string::npos)

outputFile << "Знайдено слово 'модуль'" << endl << endl;

else

outputFile << "Cлово 'модуль' - не знайдено" << endl << endl;

if (word3 != string::npos)

outputFile << "Знайдено слово 'студент'" << endl << endl;

else

outputFile << "Cлово 'студент' - не знайдено" << endl << endl;

if (word4 != string::npos)

outputFile << "Знайдено слово 'програміст'" << endl << endl;

else

outputFile << "Cлово 'програміст' - не знайдено" << endl << endl;

outputFile.close();

return true;

}

//Task 10.2

int numbersInText(string inputFileName, string inputTextInFile)

{

ofstream inputTxtFile(inputFileName, ios::app);

char numbers[10] = {'0','1','2','3','4','5','6','7','8','9'};

int numbersAmount = 0;

size\_t foundNumbers;

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

for (int j = 0; j < 10; j++){

if (inputTextInFile[i] == numbers[j]) {

numbersAmount++;

}

}

}

inputTxtFile << "Кількість цифр у вхідному файлі: "

<< numbersAmount << endl << endl;

inputTxtFile.close();

return numbersAmount;

}

bool timeInFile(string inputFileName)

{

ofstream inputFile(inputFileName, ios::app);

time\_t indefTime;

time(&indefTime);

inputFile << "Дата та час дозапису інформаціїї: "

<< ctime(&indefTime) << endl << endl;

inputFile.close();

return true;

}

//Task 10.3

double sCalculationInFile(string outputFileName, float x, float y, float z)

{

ofstream outputFile("outputFile.txt", ios::app);

outputFile << "Результат виконання функції s\_calculation: "

<< s\_calculation(x, y, z) << endl << endl;

outputFile.close();

return s\_calculation(x, y, z);

}

bool numberBInFile(string outputFileName, unsigned int b)

{

ofstream outputFile(outputFileName, ios::app);

bitset<32> b\_number(b);

outputFile << "Число " << b << " у двійковому коді: "

<< b\_number << endl;

outputFile.close();

return true;}

**Лістинг TestDriver**

#include <iostream>

#include <conio.h>

#include <cstring>

#include <clocale>

#include "ModulesNazarevskyi.h"

using namespace std;

int main()

{

char \*locale = setlocale(LC\_ALL, "ukr");

const string inputText[5] = { "Цей студент помив руки 3 рази.",

"8 Березня - це жіночий день, студентка була задоволена.",

"Я успішно виконав 3 завдання, модульчик вдався.",

"Моя програма працює успішно.",

"Цей програміст налічив 252 країни, але з них лише 195 держав"

};

const int numberSymbols[5] = {25, 47, 41, 25, 52};

const int numbersAmount[5] = {1, 2, 4, 2, 6};

int cX[5] = { 3, -5, 0, 11, -3 };

int cY[5] = { 2, -1, 12, 5, -25 };

int cZ[5] = { 5, 8, 0, -6, -7 };

double expectedResult[5] = { -3.05897, -7.1874, 0, -5.72915, -6.51101 };

const unsigned int natNumber[5] = { 5238, 27, 422, 156, 5238 };

for(int i = 0; i < 5; i++){

canToRedaction("outputFile.txt", "inputFile.txt");

cout << boolalpha

<< "10 - Запис тексту до вхідного файлу: \n"

<< inputTxtFile("inputFile.txt", inputText[i]) << endl;

cout << boolalpha

<< "10.1(1) - Запис авторської інформації до вихідного файлу: \n"

<< myInfo("outputFile.txt") << endl;

cout << boolalpha

<< "10.1(2) - Результат підрахунку кількості цифр та дозапис до вихідного файлу: \n"

<< (numbersInText("inputFile.txt", inputText[i]) == numbersAmount[i]) << endl;

cout << boolalpha

<< "10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: \n"

<< wordsInText("outputFile.txt", inputText[i]) << endl;

cout << boolalpha

<< "10.2(1) - Результат підрахунку кількості символів та дозапис до вхідного файлу: "

<< (numberSymbolsInText("inputFile.txt", inputText[i]) == numberSymbols[i]) << endl;

cout << boolalpha

<< "10.2(2) - Дозапис часу редагування файлу до вхідного файлу: \n"

<< timeInFile("inputFile.txt") << endl;

cout << boolalpha

<< "10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: \n"

<<(sCalculationInFile("outputFile.txt", cX[i], cY[i], cZ[i]) <= expectedResult[i] + 0.0005) << endl;

cout << boolalpha

<< "10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: \n"

<< numberBInFile("outputFile.txt", natNumber[i]) << endl << endl << endl;

\_getch();

}

return 0;

}

**ДОДАТОК Б**

**Модульний тест до задачі 10.1**

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_10-1 |
| Date: 04/14/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Назаревський Максим |
|  | Виконавець /  **Implementer** | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| TS\_1 | 1. Запис “Цей студент помив руки 3 рази.” у inputFile.txt  2. Запис кількості символів у inputFile.txt  3. Запис інформації у outputFile.txt  4. Запис пошуку слів у outputFile.txt | **Вміст inputFile.txt:**  Цей студент помив руки 3 рази.  Кількість символів з вхідного файлу: 25  **Вміст outputFile.txt:**  Виконавець: Назаревський Максим Володимирович  Рік розробки: 2022  Країна, місто: Україна, Кропивницький  ВНЗ: Центральний Національний Технічний Університет  Cлово 'програма' - не знайдено  Cлово 'модуль' - не знайдено  Знайдено слово 'студент'  Cлово 'програміст' - не знайдено | Passed |
| TS\_2 | 1. Запис “8 Березня - це жіночий день, студентка була задоволена.” у inputFile.txt  2. Запис кількості символів у inputFile.txt  3. Запис інформації у outputFile.txt  4. Запис пошуку слів у outputFile.txt | **Вміст inputFile.txt:**  8 Березня - це жіночий день, студентка була задоволена.  Кількість символів з вхідного файлу: 47  **Вміст outputFile.txt:**  (інформація про виконавця)  Cлово 'програма' - не знайдено  Cлово 'модуль' - не знайдено  Cлово 'студент' - не знайдено  Cлово 'програміст' - не знайдено | Passed |
| TS\_3 | 1. Запис “Я успішно виконав 3 завдання, модульчик вдався.” у inputFile.txt  2. Запис кількості символів у inputFile.txt  3. Запис інформації у outputFile.txt  4. Запис пошуку слів у outputFile.txt outputFile.txt | **Вміст inputFile.txt:**  Я успішно виконав 3 завдання, модульчик вдався.  Кількість символів з вхідного файлу: 41  **Вміст outputFile.txt:**  (інформація про виконавця)  Cлово 'програма' - не знайдено  Cлово 'модуль' - не знайдено  Cлово 'студент' - не знайдено  Cлово 'програміст' - не знайдено | Passed |
| TS\_4 | 1. Запис “Моя програма працює успішно. ” у inputFile.txt  2. Запис кількості символів у inputFile.txt  3. Запис інформації у outputFile.txt  4. Запис пошуку слів у outputFile.txt | **Вміст inputFile.txt:**  Моя програма працює успішно.  Кількість символів з вхідного файлу: 25  **Вміст outputFile.txt:**  (інформація про виконавця)  Знайдено слово 'програма'  Cлово 'модуль' - не знайдено  Cлово 'студент' - не знайдено  Cлово 'програміст' - не знайдено | Passed |
| TS\_5 | 1. Запис “Цей програміст налічив 252 країни, але з них лише 195 держав” у inputFile.txt  2. Запис кількості символів у inputFile.txt  3. Запис інформації у outputFile.txt  4. Запис пошуку слів у outputFile.txt | **Вміст inputFile.txt:**  Цей програміст налічив 252 країни, але з них лише 195 держав  Кількість символів з вхідного файлу: 52  **Вміст outputFile.txt:**  (інформація про виконавця)  Cлово 'програма' - не знайдено  Cлово 'модуль' - не знайдено  Cлово 'студент' - не знайдено  Знайдено слово 'програміст' | Passed |

**Модульний тест до задачі 10.2**

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_10-2 |
| Date: 04/14/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Назаревський Максим |
|  | Виконавець /  **Implementer** | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| TS\_1 | 1. Запис кількості цифр у inputFile.txt  2. Запис часу дозапису у inputFile.txt | **Вміст inputFile.txt:**  Цей студент помив руки 3 рази.  Кількість цифр у вхідному файлі: 1  Дата та час дозапису інформаціїї:[date] | Passed |
| TS\_2 | 1. Запис кількості цифр у inputFile.txt  2. Запис часу дозапису у inputFile.txt | **Вміст inputFile.txt:**  8 Березня - це жіночий день, студентка була задоволена.  Кількість цифр у вхідному файлі: 1  Дата та час дозапису інформаціїї:[date] | Passed |
| TS\_3 | 1. Запис кількості цифр у inputFile.txt  2. Запис часу дозапису у inputFile.txt | **Вміст inputFile.txt:**  Я успішно виконав 3 завдання, модульчик вдався.  Кількість цифр у вхідному файлі: 1  Дата та час дозапису інформаціїї:[date] | Passed |
| TS\_4 | 1. Запис кількості цифр у inputFile.txt  2. Запис часу дозапису у inputFile.txt | **Вміст inputFile.txt:**  Моя програма працює успішно.  Кількість цифр у вхідному файлі: 0  Дата та час дозапису інформаціїї:[date] | Passed |
| TS\_5 | 1. Запис кількості цифр у inputFile.txt  2. Запис часу дозапису у inputFile.txt | **Вміст inputFile.txt:**  Цей програміст налічив 252 країни, але з них лише 195 держав  Кількість цифр у вхідному файлі: 6  Дата та час дозапису інформаціїї:[date] | Passed |

**Модульний тест до задачі 10.3**

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_10-3 |
| Date: 04/14/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Назаревський Максим |
|  | Виконавець /  **Implementer** | Назаревський Максим |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| TS\_1 | 1. Запис результату s\_calculation (3, 2, 5) у outputFile.txt  2. Запис двійкового числа 5 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання функції s\_calculation: -3.05897  Число 5 у двійковому коді: 00000000000000000000000000000101 | Passed |
| TS\_2 | 1. Запис результату s\_calculation (-5, -1, 8) у outputFile.txt  2. Запис двійкового числа 27 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання функції s\_calculation: -7.1874  Число 27 у двійковому коді: 00000000000000000000000000011011 | Passed |
| TS\_3 | 1. Запис результату s\_calculation (0, 12, 0) у outputFile.txt  2. Запис двійкового числа 422 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання функції s\_calculation: 0  Число 422 у двійковому коді: 00000000000000000000000110100110 | Passed |
| TS\_4 | 1. Запис результату s\_calculation (11, 5, -6) у outputFile.txt  2. Запис двійкового числа 156 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання функції s\_calculation: -5.72915  Число 156 у двійковому коді: 00000000000000000000000010011100 | Passed |
| TS\_5 | 1. Запис результату s\_calculation (-3, 25, -7) у outputFile.txt  2. Запис двійкового числа 5238 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання функції s\_calculation: -6.51101  Число 5238 у двійковому коді: 00000000000000000001010001110110 | Passed |