МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

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

КБ 22-2

\_\_\_\_\_\_\_\_\_ Ткаченко О. С.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ Олександр СОБІНОВ

Кропивницький – 2023

**Мета роботи**

Полягає у набутті ґрунтовних вмінь та практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьної інформації.

**Завдання до лабораторної роботи**

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 програмних модулів розв’язування задач 10.1–10.3

**Варіант 5**

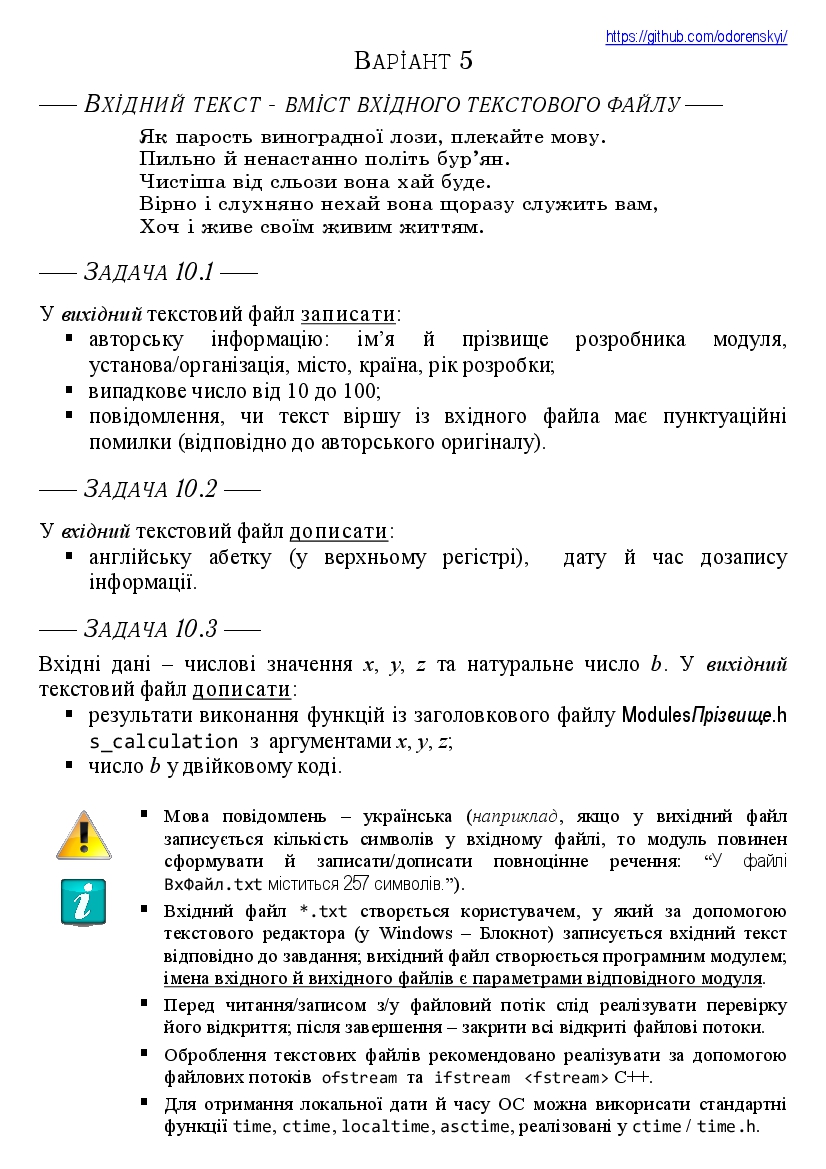


Рисунок 1 - Завдання

**Хід роботи**

На початку було завантажено Git-репозиторій і отримано завдання за варіантом. Далі в \Lab10 було заповнено README.md файл і створено теки prj, Software, TestSuite та Report.

**Розв’язання**

Згідно завдання було виконано аналіз задачі 10.1:

Вхідні данні:

* Немає;

Вихідні данні:

* Файл output.txt повинен містити персональні дані, рандомний номер від 10 до 100 і інформацію чи є пунктуаційні помилки у вірші з файлу input.txt;

Далі було створено алгоритм задачі та записано його вербальним способом:

*ПОЧАТОК*

1. Відкриваємо файл output.txt для запису;
2. Записуємо у файл output.txt персональну інформацію;
3. Генеруємо рандомне число від 10 до 100 і записуємо його у output.txt;
4. Відкриваємо файл input.txt для читання;
5. Перевіряємо чи є помилкиу вірші, і записуємо у output.txt чи є помилки;
6. Закриваємо файли output.txt і input.txt;

*КІНЕЦЬ*

Згідно завдання було виконано аналіз задачі 10.2:

Вхідні данні:

* Немає;

Вихідні данні:

* Файл input.txt повинен містити вірш, англійську абетку у верхньому регістрі і час дозапису;

Далі було створено алгоритм задачі та записано його вербальним способом:

*ПОЧАТОК*

1. Відкриваємо файл input.txt для дозапису;
2. Дозаписуємо англійську абетку у верхньому регістрі;
3. Дозаписуємо поточний час;
4. Закриваємо файл input.txt;

*КІНЕЦЬ*

Згідно завдання було виконано аналіз задачі 10.3:

Вхідні данні:

* x, y, z – числа для функції s\_calculation;
* b – натуральне число;

Вихідні данні:

* Файл output.txt повинен містити персональні дані, рандомний номер від 10 до 100 і інформацію чи є пунктуаційні помилки у вірші з файлу input.txt, та результат функції s\_calculation і число b в двійковій системі;

Далі було створено алгоритм задачі та записано його вербальним способом:

*ПОЧАТОК*

1. Відкриваємо файл output.txt для дозапису;
2. Дозаписуємо у файл output.txt результат функції s\_calculation;
3. Дозаписуємо у output.txt число b в двійковій формі;
4. Закриваємо файл output.txt;

*КІНЕЦЬ*

Потім було розроблено TestSuite для задач 10.1-10.3 – Додаток C – TestSuite.

Лістинг ModulesTkachenko:

# define \_USE\_MATH\_DEFINES

#include <math.h>

#include <iostream>

#include <fstream>

#include <ctime>

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

{

float S = log(abs(sin(z))) + (0.5 \* pow(x,2) - sqrt(abs(pow(y+z,2) - pow(x,5)))) / 10 \* M\_PI \* pow(z,4);

return S;

}

std::string function\_9\_1 (int ball\_bofort)

{

switch (ball\_bofort)

{

case 0:

return "Швидкість вітру: < 0,0003 км/с. Дія вітру: відсутність вітру; дим піднімається прямовисто; листя дерев нерухомі.";

case 1:

return "Швидкість вітру: 0,0003 - 0,0015 км/с. Дія вітру: дим 'пливе'; флюгер не обертається.";

case 2:

return "Швидкість вітру: 0,0016 - 0,0024 км/с. Дія вітру: рух повітря відчувається обличчям; шелестить листя; флюгер обертається.";

case 3:

return "Швидкість вітру: 0,0024 - 0,0054 км/с. Дія вітру: тріпоче листя, хитаються дрібні гілки; майорять прапори.";

case 4:

return "Швидкість вітру: 0,0055 - 0,0079 км/с. Дія вітру: хитаються тонкі гілки; вітер піднімає пил та шматки паперу.";

case 5:

return "Швидкість вітру: 0,008 - 0,0107 км/с. Дія вітру: хитаються великі гілки; на воді з'являються хвилі.";

case 6:

return "Швидкість вітру: 0,0108 - 0,0138 км/с. Дія вітру: хитаються великі гілки.";

case 7:

return "Швидкість вітру: 0,0139 - 0,0171 км/с. Дія вітру: хитаються невеликі стовбури дерев; на морі здіймаються хвилі, піняться.";

case 8:

return "Швидкість вітру: 0,0172 - 0,0207 км/с. Дія вітру: ламаються гілки дерев, і важко йти проти вітру.";

case 9:

return "Швидкість вітру: 0,0208 - 0,0244 км/с. Дія вітру: невеликі руйнування; зриває покрівлі, руйнує димарі.";

case 10:

return "Швидкість вітру: 0,0245 - 0,0284 км/с. Дія вітру: значні руйнування; дерева вириваються з корінням.";

case 11:

return "Швидкість вітру: 0,0285 - 0,0326 км/с. Дія вітру: великі руйнування.";

case 12:

return "Швидкість вітру: >= 0,0327 км/с. Дія вітру: призводить до спустошень.";

default:

return "Error: incorrect input";

}

}

std::string function\_9\_2 (int girth\_of\_head)

{

if (girth\_of\_head >= 480 && girth\_of\_head <= 490)

{

return "Розмір: S, дитячий";

}

else if (girth\_of\_head >= 500 && girth\_of\_head <= 510)

{

return "Розмір: M, дитячий";

}

else if (girth\_of\_head >= 520 && girth\_of\_head <= 530)

{

return "Розмір: L, дитячий";

}

else if (girth\_of\_head >= 530 && girth\_of\_head <= 540)

{

return "Розмір: XS, дорослий";

}

else if (girth\_of\_head >= 550 && girth\_of\_head <= 560)

{

return "Розмір: S, дорослий";

}

else if (girth\_of\_head >= 570 && girth\_of\_head <= 580)

{

return "Розмір: M, дорослий";

}

else if (girth\_of\_head >= 590 && girth\_of\_head <= 600)

{

return "Розмір: L, дорослий";

}

else if (girth\_of\_head >= 610 && girth\_of\_head <= 620)

{

return "Розмір: XL, дорослий";

}

else

{

return "Error: incorrect input";

}

}

std::string function\_9\_3 (int N)

{

if (N < 0 || N > 8999)

return "Error: incorrect input";

int bit\_of\_N = (N >> 10) & 1;

if (bit\_of\_N == 0)

{

int amount = 0;

while ((N & 1) == 0)

{

amount++;

N >>= 1;

}

std::string result = "Кількість двійкових нулів: " + std::to\_string(amount);

return result;

}

else

{

int amount = 0;

while (N != 0)

{

if (N & 1)

amount++;

N >>= 1;

}

std::string result = "Кількість двійкових одиниць: " + std::to\_string(amount);

return result;

}

}

void function\_10\_1()

{

fstream output;

output.open("D:\\Леха КИСМ\\БМТП\\2 семестр\\Лабы\\Tkachenko-Oleksii-KB222\\lab10\\prj\\output.txt", ios\_base::out);

output << "Олексій Ткаченко, ЦНТУ, Кропивницький, Україна, 2023" << endl;

int random\_number;

srand(time(NULL));

random\_number = rand() % 100 + 10;

if (random\_number > 100)

output << random\_number - 10 << endl;

else

output << random\_number << endl;

fstream input;

input.open("D:\\Леха КИСМ\\БМТП\\2 семестр\\Лабы\\Tkachenko-Oleksii-KB222\\lab10\\prj\\input.txt", ios\_base::in);

string line;

string original\_text [5] =

{

"Як парость виноградної лози, плекайте мову.",

"Пильно й ненастанно політь бур'ян.",

"Чистіша від сльози вона хай буде.",

"Вірно і слухняно нехай вона щоразу служить вам,",

"Хоч і живе своїм живим життям."

};

int i = 0;

int j = 0;

while (getline(input, line))

{

if(line == original\_text[i])

j += 1;

if (i == 4)

{

if (j == 5)

output << "Пунктуаційних помилок немає" << endl;

else

output << "Пунктуаційні помилки є" << endl;

}

i += 1;

}

input.close();

output.close();

}

void function\_10\_2()

{

fstream f;

f.open("D:\\Леха КИСМ\\БМТП\\2 семестр\\Лабы\\Tkachenko-Oleksii-KB222\\lab10\\prj\\input.txt", ios\_base::app);

f << "\nABCDEFGHIJKLMNOPQRSTUVWXYZ\n";

time\_t rawtime;

time(&rawtime);

f << ctime(&rawtime);

f.close();

}

void function\_10\_3(float x, float y, float z, int b)

{

fstream f;

f.open("D:\\Леха КИСМ\\БМТП\\2 семестр\\Лабы\\Tkachenko-Oleksii-KB222\\lab10\\prj\\output.txt", ios\_base::app);

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

for (int i = 31; i >= 0; i--)

{

if (b & (1 << i))

f << "1";

else

f << "0";

}

f << endl;

f.close();

}

Лістинг TestDriver:

# include <iostream>

#include <fstream>

#include <ctime>

#include <locale>

#include "ModulesTkachenko.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "ukr");

function\_10\_1();

fstream output;

output.open("C:\\Users\\User\\Documents\\лаб10\\output.txt", ios\_base::in);

fstream input;

input.open("C:\\Users\\User\\Documents\\лаб10\\input.txt", ios\_base::in);

string line;

int number;

string output\_text [2] = {"Олексій Ткаченко, ЦНТУ, Кропивницький, Україна, 2023", "Пунктуаційних помилок немає"};

string result;

int i = 0;

int j = 0;

while (getline(output, line))

{

if (i == 0)

{

if(line == output\_text[0])

j += 1;

}

if (i == 1)

{

number = atoi(line.c\_str());

if (number>=10 &&number <=100)

j += 1;

}

if (i == 2)

{

if(line == output\_text[1])

j += 1;

}

i += 1;

if (i==3)

{

if(j==3)

cout <<"Function\_10\_1 output is passed" << endl;

else

cout << "Function\_10\_1 output is failed" << endl;

}

}

string input\_text [7] =

{

"Як парость виноградної лози, плекайте мову.",

"Пильно й ненастанно політь бур'ян.",

"Чистіша від сльози вона хай буде.",

"Вірно і слухняно нехай вона щоразу служить вам,",

"Хоч і живе своїм живим життям.",

"",

"ABCDEFGHIJKLMNOPQRSTUVWXYZ"

};

i = 0;

j = 0;

while (getline(input, line))

{

if(line == input\_text[i])

j += 1;

if (i == 4)

{

if (j == 5)

cout << "Function\_10\_1 input is passed" << endl;

else

cout << "Function\_10\_1 input is failed" << endl;

}

i += 1;

if (i == 5)

break;

}

input.close();

output.close();

function\_10\_2();

input.open("C:\\Users\\User\\Documents\\лаб10\\input.txt", ios\_base::in);

string time1;

time\_t rawtime;

time(&rawtime);

time1 = ctime(&rawtime);

i = 0;

j = 0;

while (getline(input, line))

{

if (line == input\_text[i])

j += 1;

if (line == time1)

{

if (j == 7)

cout << "Function\_10\_2 is passed" << endl;

else

cout << "Function\_10\_2 is failed" << endl;

}

i += 1;

}

input.close();

function\_10\_3(1, 2, 5, 6);

output.open("C:\\Users\\User\\Documents\\лаб10\\output.txt", ios\_base::in);

string text [2] = {"-1262.22", "00000000000000000000000000000110"};

i = 0;

j = 0;

while (getline(output, line))

{

if (line == text[0] || line == text[1])

j += 1;

if (i == 4)

{

if (j == 2)

cout << "Function\_10\_3 is passed" << endl;

else

cout << "Function\_10\_3 is failed" << endl;

}

i += 1;

}

output.close();

return 0;

}

Результат роботи TestDriver:

Function\_10\_1 output is passed

Function\_10\_1 input is passed

Function\_10\_2 is passed

Function\_10\_3 is passed

**Висновки**

Під час підготовки до виконання лабораторної роботи було належно опрацьовано рекомендовану літературу та контрольні запитання. Після ознайомлення з порядком проведення лабораторної роботи, було розпочато її виконання.

Згідно завдання, було завантажено Git-репозиторій і в \Lab10 було створено теки prj, Software, TestSuite та Report.

Потім було виконано аналіз умови і постановку задач 10.1-10.3, на основі чого було розроблено TestSuite10.1, TestSuite10.2, TestSuite10.3. В Code::Blocks IDE було відкрито проект статичної бібліотеки ModulesTkachenko з \Lab8\prj. Потім на основі результатів проектування програмних модулів, мовою С++ було виконано конструювання функцій, які реалізовують розв’язування задач 10.1-10.3. Далі проект було скомпільовано.

Потім я відкрив проект заголовкового файлу ModulesTkachenko з \Lab8\prj та доповнив його прототипами функцій розв’язування задач 10.1-10.3.

Потім в Code::Blocks у теці \Lab10\prj було створено проект консольного додатка TestDriver, реалізувано тестовий драйвер і ппроведено модульне тестування функцій розв’язування задач 10.1-103. TestDriver.exe було скопійовано в \Software. Після цього за допомогою TestDriver.ехе було виконано автоматизоване тестування розроблених функцій розв’язування задач 10.1–10.3. і на цьому виконання лабораторної роботи було закінчено.

Процес виконання лабораторної роботи був цікавим.

**Додаток А - Test Suite**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TestSuite |
| Рівень тестування  Level of Testing | автоматизований / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Ткаченко Олексій Сергійович |
| Виконавець  Implementer | Ткаченко Олексій Сергійович |

Artifact: Test Suite

Date: 4/21/2023

|  |  |  |  |
| --- | --- | --- | --- |
| **10.1** | | | |
| Preliminary Steps | Action (test steps) | Expected Result | Result |
| Вихідний файл output.txt:  Вхідний файл input.txt:  *Як парость виноградної лози, плекайте мову.*  *Пильно й ненастанно політь бур'ян.*  *Чистіша від сльози вона хай буде.*  *Вірно і слухняно нехай вона щоразу служить вам,*  *Хоч і живе своїм живим життям.* | Викликаємо function\_10\_1() | Вихідний файл output.txt:  *Олексій Ткаченко, ЦНТУ, Кропивницький, Україна, 2023*  *(число від 10 до 100)*  *Пунктуаційних помилок немає*  Вхідний файл input.txt:  *Як парость виноградної лози, плекайте мову.*  *Пильно й ненастанно політь бур'ян.*  *Чистіша від сльози вона хай буде.*  *Вірно і слухняно нехай вона щоразу служить вам,*  *Хоч і живе своїм живим життям.* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **10.2** | | | |
| Preliminary Steps | Action (test steps) | Expected Result | Result |
| Вихідний файл input.txt:  *Як парость виноградної лози, плекайте мову.*  *Пильно й ненастанно політь бур'ян.*  *Чистіша від сльози вона хай буде.*  *Вірно і слухняно нехай вона щоразу служить вам,*  *Хоч і живе своїм живим життям.* | Викликаємо function\_10\_2() | Вихідний файл input.txt:  *Як парость виноградної лози, плекайте мову.*  *Пильно й ненастанно політь бур'ян.*  *Чистіша від сльози вона хай буде.*  *Вірно і слухняно нехай вона щоразу служить вам,*  *Хоч і живе своїм живим життям.*  *ABCDEFGHIJKLMNOPQRSTUVWXYZ*  *(дата в момент запису)* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **10.3** | | | |
| Preliminary Steps | Action (test steps) | Expected Result | Result |
| Вихідний файл output.txt:  *Олексій Ткаченко, ЦНТУ, Кропивницький, Україна, 2023*  *(число від 10 до 100)*  *Пунктуаційних помилок немає* | Викликаємо function\_10\_3(1, 2, 5, 6) | Вихідний файл output.txt:  *Олексій Ткаченко, ЦНТУ, Кропивницький, Україна, 2023*  *(число від 10 до 100)*  *Пунктуаційних помилок немає*  *-1262.22*  *00000000000000000000000000000110* |  |