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”

на тему

“Реалізація програмних модулів оброблення даних складових типів з файловим введенням / виведенням”

**Виконав**

студент академічної

групи КБ-21

\_\_\_\_\_\_\_\_\_  Е.В. Ісаченков

**Приймає**

викладач кафедри

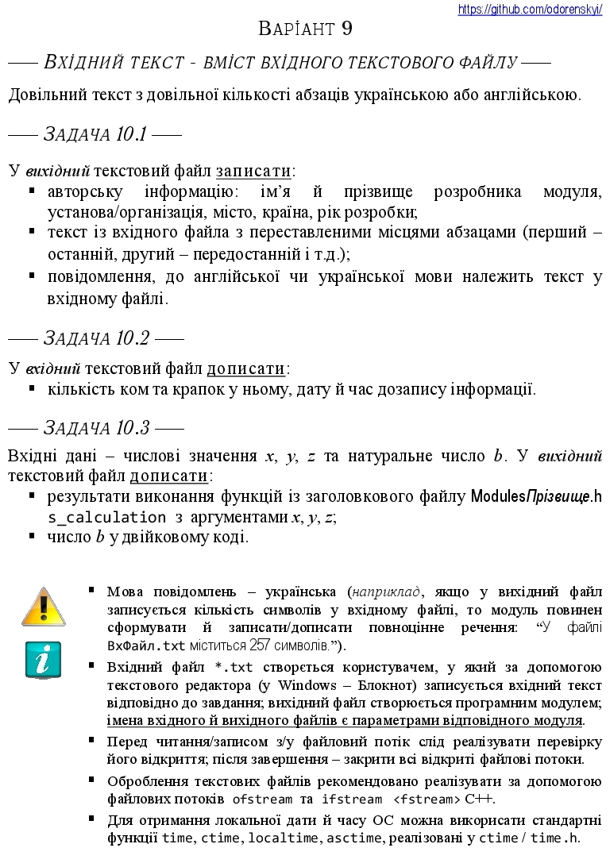
кібербезпеки та ПЗ

\_\_\_\_\_\_\_\_\_  П.С. Усік

Кропивницький – 2022

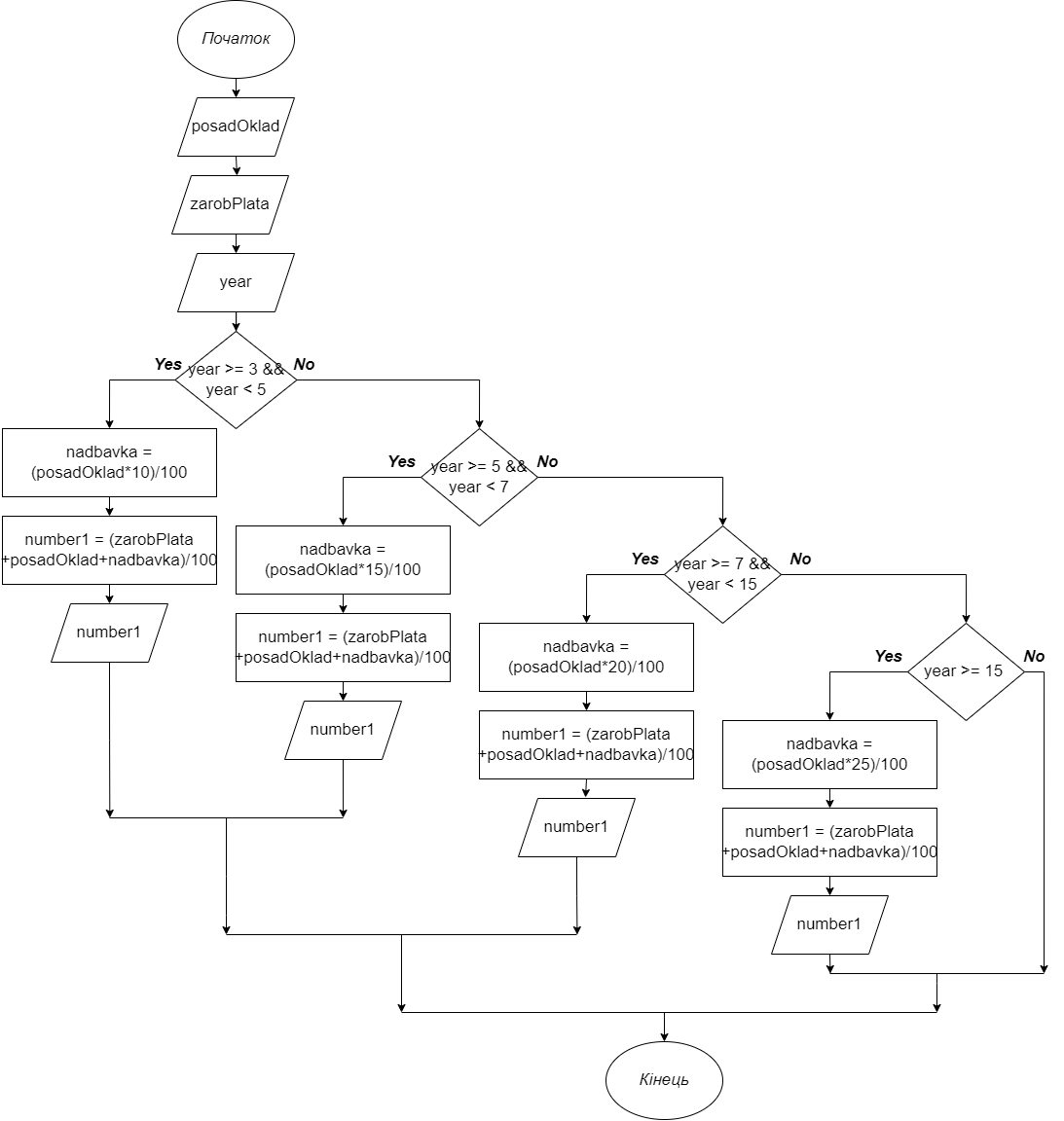
**Лабораторна робота №10**

*Мета роботи* полягає у набутті ґрунтовних вмінь і 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.

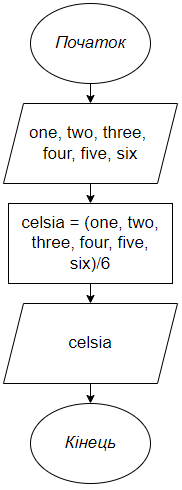
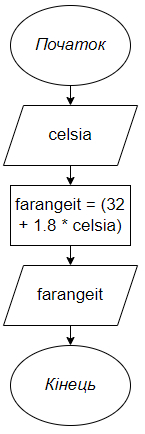
****

***Алгоритми виконання:***

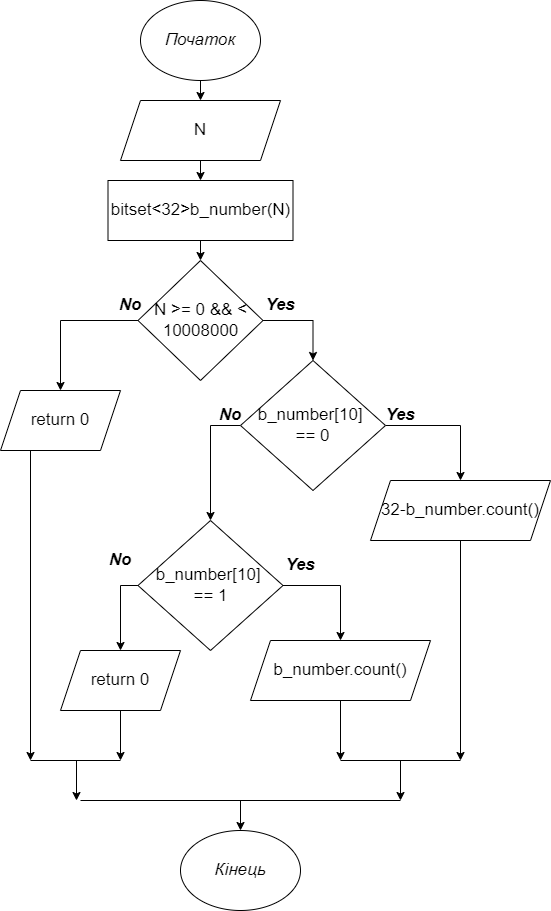
***Алгоритм виконання Task 9.1:***

******

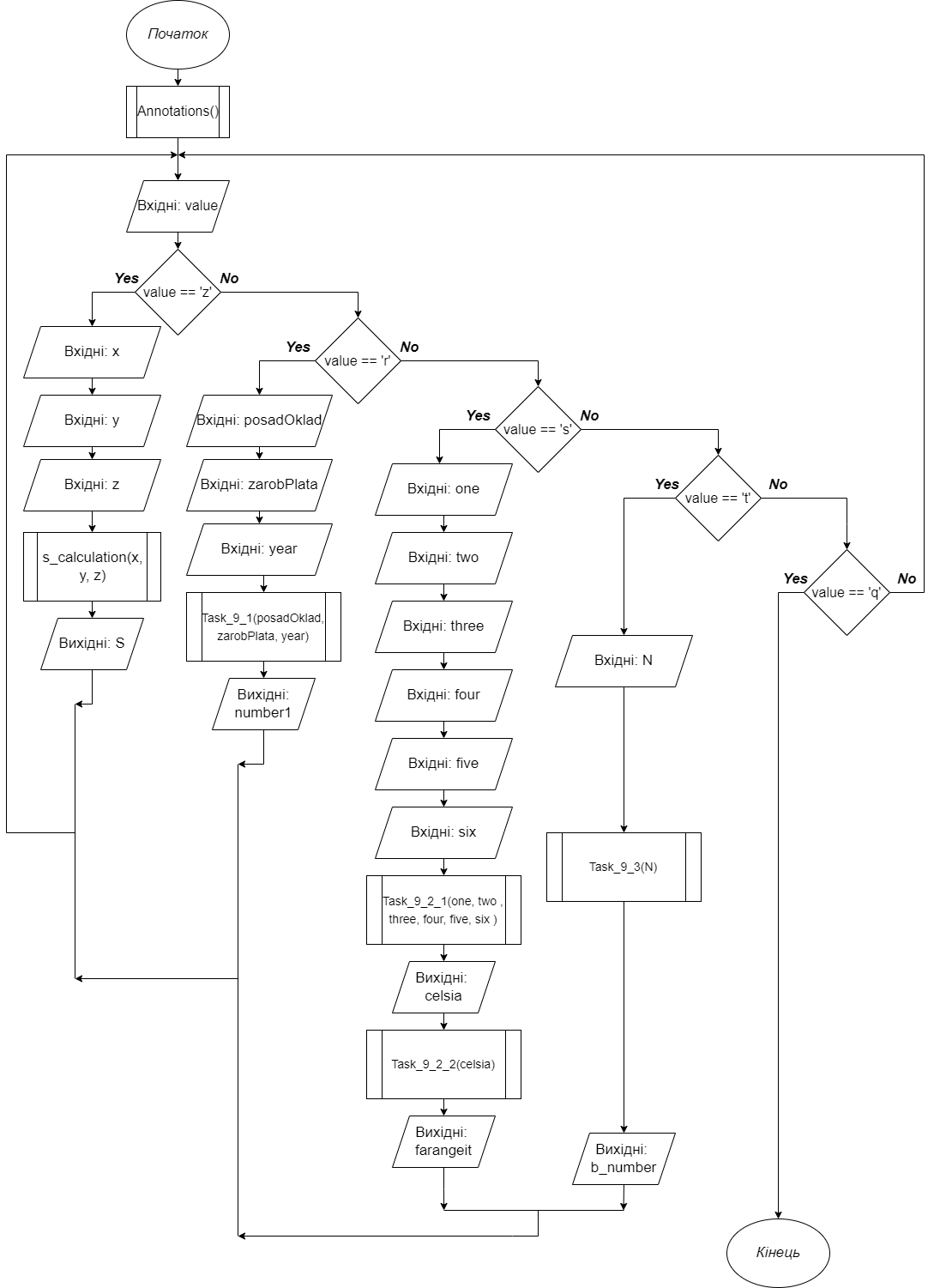
***Алгоритм виконання Task 9.2:***

*** ***

***Алгоритм виконання Task 9.3:***

******

***Алгоритм виконання Task 9.4:***

**

***Висновок:***

Мета цієї роботи полягає у набутті 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.

Для виконання даної лабораторної роботи потрібно приблизно – 4 академічні години.

Також потрібно таке обладнення:

* персональний комп’ютер з операційною системою Windows;
* вільне кросплатформове Code::Blocks IDE ([www.codeblocks.org](http://www.codeblocks.org));
* текстовий редактор (OpenOffice Writer, Microsoft Word або ін.);
* – файл-шаблон тестового набору Artifact\_TEST\_SUITE\_lab.doc;
* власні обліковий запис на GitHub https://github.com/ та Git-репозиторій <https://github.com/odorenskyi/Isachenkov-Eduard-KB-21>

При підготовці до лабораторної роботи я отримав такі завдання.

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).
2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а. Я працював за таким порядком виконання лабораторної роботи:

Я дотримувався такого плану виконання лабораторної роботи №9:

1. Спочатку я зайшов на мій Git Repositories та отримав завдання для виконання лабораторної роботи №9.
2. Завантажив Git Repositories на мій диск.
3. Змінив вміст файлу *README.md*, вказавши: тему, мету, варіант та завдання 9.1-9.4.
4. Створив теки: prj, SoftWare, TestSuite, Report. Також завантажив все на репозиторій.
5. Здійснив аналіз задач 9.1 – 9.3.
6. Створив блок-схему до неї та розробив набір контрольних прикладів до задачі 9.1 – 9.3, задля виконання модульного тестування модулів С++.
7. З лабораторної роботи №8, перемістив статичну бібліотеку *ModulesIsachenkov* в папку prj.
8. Реалізував функції: *Task\_9\_1, Task\_9\_2, Task\_9\_3*.
9. Скомпілюв проєкт статичної бібліотеки.
10. З лабораторної роботи №8, перемістив заголовковий файл *ModulesIsachenkov.h*, в ньому вписав прототипи функцій *Task\_9\_1, Task\_9\_2, Task\_9\_3.*
11. *У* Code::Blocks створив проєкт консольного застосунку С++, іменував його як *TestDriver*.
12. Реалізував мовою програмування С++, тестовий драйвер, підключив статичну бібліотеку та заголовковий файл в налаштуваннях компілятора.
13. Здійснив аналіз задачі 9.4
14. Створив блок-схему до неї. Завантажив тест-сьют, та заповнив його належним чином. Для повної перевірки проєкта потрібно 6 тест-кейсів.
15. Створив у Code::Blocks проєкт консольного застосунку під назвою Isachenkov-task.
16. Написав лістинг до завдання 9.4.
17. Зробив перевірку за допомогою системного тестування, відповіді записав у TestSuite.
18. Надіслав всі файли на Git Repositorie.

Данна лабораторна робота розширила мої знання. Вона була чудова!

***Додаток №1:***

***Вхідні та вихідні дані до TestDriver:***

|  |  |  |
| --- | --- | --- |
| **ID TS** | **Вхідні**  **дані** | **Вихідні**  **дані** |
| **№** | **Завдання 9.1** | |
| **1** | 1000, 5000, 9 | 6200 |
| **2** | 5000, 10000, 4 | 12750 |
| **3** | 3000, 15450, 16 | 19200 |
| **№** | **Завдання 9.2** | |
| **1** | 1, 4, 8, 10, 15, 21 | 9 |
| **2** | -6, -2, 1, 4, 8, 12 | 2 |
| **3** | 1, 8, 16, 22, 26, 34 | 17 |
| **№** | **Завдання 9.3** | |
| **1** | 1234567 | 11 |
| **2** | 7654321 | 19 |
| **3** | 57210 | 12 |

***Додаток результат виконання TestDriver (з консольного вікна):***

*Test Task\_9\_1*

*Test case #01 PASSED.*

*Test case #02 PASSED.*

*Test case #03 PASSED.*

*Test Task\_9\_2*

*Test case #01 PASSED.*

*Test case #02 PASSED.*

*Test case #03 PASSED.*

*Test Task\_9\_3*

*Test case #01 PASSED.*

*Test case #02 PASSED.*

*Test case #03 PASSED.*

***Додаток №2:***

***Вихідний код проєкту ModulesIsachenkov:***

#include <cmath>

#include <iostream>

#include <bitset>

#include <fstream>

#include <locale.h>

#include <Windows.h>

#include <ctime>

using namespace std;

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

return (pow((2 \* z + 1), x)) - (sqrt(abs(y - 0.5 \* z))) + z + 3.141592653589793;

}

int Task\_9\_1(int posadOklad, int zarobPlata, int year){

int nadbavka, number1;

if (year >= 3 && year < 5){

nadbavka = (posadOklad\*10)/100;

number1 = zarobPlata+posadOklad+nadbavka;

}

if (year >= 5 && year < 7){

nadbavka = (posadOklad\*15)/100;

number1 = zarobPlata+posadOklad+nadbavka;

}

if (year >= 7 && year < 15){

nadbavka = (posadOklad\*20)/100;

number1 = zarobPlata+posadOklad+nadbavka;

}

if (year >= 15){

nadbavka = (posadOklad\*25)/100;

number1 = zarobPlata+posadOklad+nadbavka;

}

return number1;

}

int Task\_9\_2\_1(int one, int two, int three, int four, int five, int six){

int celsia;

celsia = (one+two+three+four+five+six)/6;

return celsia;

}

int Task\_9\_2\_2(int celsia){

int farangeit;

farangeit = (32 + 1.8 \* celsia);

return farangeit;

}

int Task\_9\_3(int N){

bitset<32>b\_number(N);

if (N >= 0 && N < 10008000){

if (b\_number[10] == 0){

return 32-b\_number.count();

}

else if (b\_number[10] == 1){

return b\_number.count();

}

}

else{

return 0;

}

}

char Task\_10\_1(char in\_path[], char out\_path[]){

string annotationsUkr = {"|---------------------------------------------------------------------------------------|\n|Ісаченков Едуард Віталійович, |\n|Центральноукраїнського національного технічного університету,|\n|Кропивницький, Україна, Рік розробки - 26.03.2022 |\n|---------------------------------------------------------------------------------------|\n"};

string annotationsEng = {"|---------------------------------------------------------------------------------------|\n|Isachenkov Eduard Vitaliyovych, |\n|Central Ukrainian National Technical University, |\n|Kropyvnytskyi, Ukraine, Year of construction - 26.03.2022 |\n|---------------------------------------------------------------------------------------|\n"};

ofstream fout;

ifstream fin;

fout.open(out\_path, ofstream::app);

fin.open(in\_path);

if(!fin.is\_open()){

cout << "Помилка вiдкривання файлу №1" << endl;

}

else{

cout << "Вiдкривання файлу пройшло успiшно! №1\n" << endl;

char ch;

string str1, str2, str3, str4;

int num1, num2, num3, k = 0;

while(fin.get(ch)){

if(ch >= 'а' && ch <= 'я') k++;

if(ch != '@') str1 += ch;

else {

num1 = fin.tellg();

break;

}

}

fin.seekg(num1, ios\_base::beg);

while(fin.get(ch)){

if(ch != '@') str2 += ch;

else {

num2 = fin.tellg();

break;

}

}

fin.seekg(num2, ios\_base::beg);

while(fin.get(ch)){

if(ch != '@') str3 += ch;

else {

num3 = fin.tellg();

break;

}

}

if(k >= 1) num3 -= 2;

fin.seekg(num3, ios\_base::beg);

while(fin.get(ch)){

if(ch != '@') str4 += ch;

else break;

}

if(k >= 1) {

fout << annotationsUkr;

fout << "Текст написаний Українською мовою\n\n";

fout << '\t';

fout << str4;

fout << '\n';

fout << '\t';

fout << str3;

fout << str2;

fout << '\n';

fout << str1;

}

else {

fout << annotationsEng;

fout << "The text is written in English\n\n";

fout << '\t';

fout << str4;

fout << '\n';

fout << '\t';

fout << str3;

fout << str2;

fout << '\n';

fout << str1;;

}

}

fin.close();

fout.close();

}

char Task\_10\_2(char in\_path[]){

ifstream fin;

ofstream fout;

time\_t rawtime;

time(&rawtime);

fin.open(in\_path);

fout.open(in\_path, ofstream::app);

if(!fin.is\_open()){

cout << "Помилка вiдкривання файлу №2 " << endl;

}

else{

cout << "Вiдкривання файлу пройшло успiшно! №2\n" << endl;

int koma, krapka, k;

char ch;

while(fin.get(ch)){

if(ch >= 'а' && ch <= 'я') k++;

if(ch == ',') koma++;

if(ch == '.') krapka++;

}

if (k >= 1) {

fout << '\n';

fout << ctime(&rawtime);

fout << '\n';

fout << "В данному тексті ком: = ";

fout << koma;

fout << '\n';

fout << "В данному тексті крапок: = ";

fout << krapka;

fout << '\n';

}

else {

fout << '\n';

fout << ctime(&rawtime);

fout << '\n';

fout << "In this text com: = ";

fout << koma;

fout << '\n';

fout << "In this text dots: = ";

fout << krapka;

fout << '\n';

}

}

fin.close();

fout.close();

}

void Task\_10\_3(int x, int y, int z, int b, char out\_path[]){

ofstream fout;

int bin = 0, k = 1;

while (b){

bin += (b % 2) \* k;

k \*= 10;

b /= 2;

}

fout.open(out\_path, ofstream::app);

fout << endl;

fout << "S: = ";

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

fout << endl;

fout << "Число b, в двійкомову коді: =";

fout << bin;

fout.close();

}

***Вихідний код проєкту TestDriver:***

#include <iostream>

#include "ModulesIsachenkov.h"

using namespace std;

void testdriver\_1(void){

int posadOklad[3] = {1000, 2500, 3000};

int zarobPlata[3] = {5000, 10000, 15450};

int year[3] = {9, 4, 16};

int num1 [3] = {6200, 12750, 19200};

cout << "Test Task\_9\_1\n";

for(int i = 0; i < 3; i++){

if (Task\_9\_1(posadOklad[i], zarobPlata[i], year[i]) == num1[i])

cout << "\nTest case #0" << i+1 << " PASSED.\n";

else

cout << "\nTest case #0" << i+1 << " FAILED.\n";

}

}

void testdriver\_2(void){

int one[3] = {1, -6, 1};

int two[3] = {4, -2, 8};

int three[3] = {8, 1, 16};

int four[3] = {10, 4, 22};

int five[3] = {15, 8, 26};

int six[3] = {21, 12, 34};

int num2\_1[3] = {9, 2, 17};

int num2\_2[3] = {49, 37, 64};

cout << "\nTest Task\_9\_2\n";

for(int i = 0; i < 3; i++){

if ((Task\_9\_2\_1(one[i], two[i], three[i], four[i], five[i], six[i]) == num2\_1[i]) && (Task\_9\_2\_2((Task\_9\_2\_1(one[i], two[i], three[i], four[i], five[i], six[i])) == num2\_2[i])))

cout << "\nTest case #0" << i+1 << " PASSED.\n";

else

cout << "\nTest case #0" << i+1 << " FAILED.\n";

}

}

void testdriver\_3(void){

int N[3] = {1234567, 7654321, 57210};

int k[3] = {11, 19, 12};

cout << "\nTest Task\_9\_3\n";

for(int i = 0; i < 3; i++){

if (Task\_9\_3(N[i]) == k[i])

cout << "\nTest case #0" << i+1 << " PASSED.\n";

else

cout << "\nTest case #0" << i+1 << " FAILED.\n";

}

}

int main()

{

testdriver\_1();

testdriver\_2();

testdriver\_3();

return 0;

}