Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КН-23

Гребенюк Д. О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

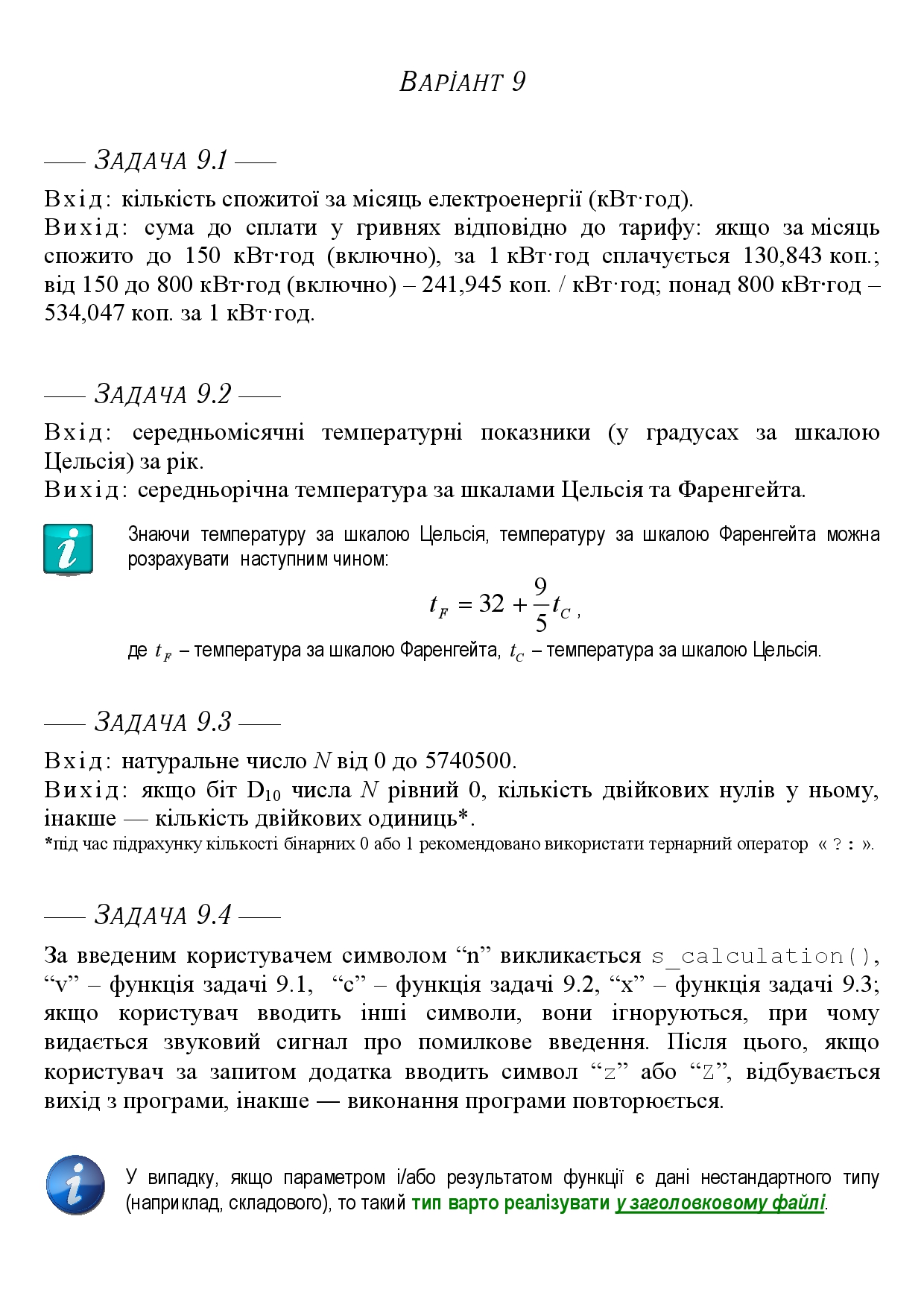
**Тема:** Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів

**Завдання до лабораторної роботи:**

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.

**Варіант 9**



**Задача 9.1**

Вхідні дані: kWh – цілочисельний тип int – кількість спожитої електроенергії за місяць (кВт**⋅**год)

Вихідні дані: sum – дробовий тип float – сума до сплати у гривнях

Прототип функції: float calculateElectricityConsume(int);

*Математична модель:*

Якщо kWh <= 150, то

sum = kWh \* 130.843

Якщо kWh > 150 && kWh <= 800, то

sum = kWh \* 241.945

Якщо kWh > 800, то

sum = kWh \* 534.047

**Задача 9.2**

Вхідні дані: averageMonthTemperature – масив цілочисельного типу int – середній показник температури за кожен місяць

Вихідні дані: averageYearTemperature – складений тип, структура (поля: float celsius, float fahrenheit) – середньорічна температура за шкалами Цельсія та Фаренгейта

Прототип функції: Temperature averageYearTemperature(int averageMonthTemperature[12]);

*Математична модель:*

1. сума температур за 12 місяців / 12 – *температура за шкалою Цельсія*;

2. 32 + (9 / 5) \* температуру за шкалою Цельсія – *температура за шкалою Фаренгейта*.

**Задача 9.3**

Вхідні дані: N – беззнаковий цілочисельний тип int – натуральне число від 0 до 5740500

Вихідні дані: count – беззнаковий цілочисельний тип int – кількість двійкових нулів або одиниць (в залежності від числа N)

Алгоритм:

*ПОЧАТОК*

1. Отримати десятий біт числа N;

2. Якщо десятий біт рівний 0, то знайти кількість двійкових нулів у числі;

3. Якщо дестий біт рівний 1, то знайти кількість двійкових одиниць у числі;

*КІНЕЦЬ*

*Реалізація завдань 9.1-9.3 знаходиться у лабораторній роботі №8, в теці prj/ModulesHrebeniuk.*

*Результат виконання тестового драйвера:*

=== TS\_9\_1 ===

Тест-кейс №1

Вхідні дані:

kWh: 130

Результат: 17009.6

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

kWh: 151

Результат: 36533.7

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

kWh: 515

Результат: 124602

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

kWh: -1

Результат: 0

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

kWh: 0

Результат: 0

Статус тест-кейса: passed

Тест-кейс №6

Вхідні дані:

kWh: 820

Результат: 437919

Статус тест-кейса: passed

Тест-кейс №7

Вхідні дані:

kWh: 155

Результат: 37501.5

Статус тест-кейса: passed

Тест-кейс №8

Вхідні дані:

kWh: 80

Результат: 10467.4

Статус тест-кейса: passed

Тест-кейс №9

Вхідні дані:

kWh: 1500

Результат: 801070

Статус тест-кейса: passed

Тест-кейс №10

Вхідні дані:

kWh: 1331

Результат: 710817

Статус тест-кейса: passed

=== TS\_9\_2 ===

Тест-кейс №1

Вхідні дані:

averageMonthTemperature: -6, 1, 20, -18, -7, 15, 25, -17, -28, 26, -30, 8

Результат:

Celsius: -0.916667

Fahrenheit: 30.35

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

averageMonthTemperature: -23, 29, -1, 4, 2, -13, 15, 1, 15, -10, 18, -15

Результат:

Celsius: 1.83333

Fahrenheit: 35.3

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

averageMonthTemperature: -6, 14, -5, -22, 12, -1, -30, -22, -29, 20, -10, -5

Результат:

Celsius: -7

Fahrenheit: 19.4

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

averageMonthTemperature: 4, -14, 10, 8, -19, 10, -13, -12, 9, 19, 23, -20

Результат:

Celsius: 0.416667

Fahrenheit: 32.75

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

averageMonthTemperature: -25, 9, 11, 22, -29, 0, -24, -3, -15, 3, 7, 30

Результат:

Celsius: -1.16667

Fahrenheit: 29.9

Статус тест-кейса: passed

Тест-кейс №6

Вхідні дані:

averageMonthTemperature: -27, 7, -23, -25, -1, -3, 0, -28, 15, -21, 13, 27

Результат:

Celsius: -5.5

Fahrenheit: 22.1

Статус тест-кейса: passed

Тест-кейс №7

Вхідні дані:

averageMonthTemperature: 22, 30, -16, 0, 18, -22, 10, 25, 20, -7, 16, 21

Результат:

Celsius: 9.75

Fahrenheit: 49.55

Статус тест-кейса: passed

Тест-кейс №8

Вхідні дані:

averageMonthTemperature: 25, 24, -11, -20, -1, 26, -21, 2, 4, -12, 9, -28

Результат:

Celsius: -0.25

Fahrenheit: 31.55

Статус тест-кейса: passed

Тест-кейс №9

Вхідні дані:

averageMonthTemperature: 17, -20, -25, 1, -8, 20, -3, -17, 19, 14, 13, 9

Результат:

Celsius: 1.66667

Fahrenheit: 35

Статус тест-кейса: passed

Тест-кейс №10

Вхідні дані:

averageMonthTemperature: 24, -6, 5, 13, 19, -9, 5, 14, -14, 24, 26, 17

Результат:

Celsius: 9.83333

Fahrenheit: 49.7

Статус тест-кейса: passed

=== TS\_9\_3 ===

Тест-кейс №1

Вхідні дані:

N: 2527920

Результат: 9

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

N: 2840999

Результат: 19

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

N: 5116662

Результат: 12

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

N: 1280412

Результат: 22

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

N: 1698360

Результат: 11

Статус тест-кейса: passed

Тест-кейс №6

Вхідні дані:

N: 413992

Результат: 25

Статус тест-кейса: passed

Тест-кейс №7

Вхідні дані:

N: 4027283

Результат: 14

Статус тест-кейса: passed

Тест-кейс №8

Вхідні дані:

N: 5401843

Результат: 19

Статус тест-кейса: passed

Тест-кейс №9

Вхідні дані:

N: 2933116

Результат: 21

Статус тест-кейса: passed

Тест-кейс №10

Вхідні дані:

N: 5156645

Результат: 13

Статус тест-кейса: passed

*Лістинг проекту ModulesHrebeniuk:*

#include <cmath>

#include "ModulesHrebeniuk.h"

using namespace std;

// Laboratory work №8

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

float S = M\_PI \* x + pow(M\_E, abs(y)) - sqrt(abs(pow(z, 2) - y));

return S;

}

// Laboratory work №9

float calculateElectricityConsume(int kWh) {

float sum = 0.0;

if (kWh > 0 && kWh <= 150) {

sum = kWh \* 130.843;

} else if (kWh > 150 && kWh <= 800) {

sum = kWh \* 241.945;

} else if (kWh > 800) {

sum = kWh \* 534.047;

}

return sum;

}

Temperature averageYearTemperature(int averageMonthTemperature[MONTH\_IN\_YEAR]) {

Temperature averageYearTemperature;

int totalMonthTemperature = 0;

for (int i = 0; i < MONTH\_IN\_YEAR; ++i) {

totalMonthTemperature += averageMonthTemperature[i];

}

averageYearTemperature.celsius = (float)totalMonthTemperature / MONTH\_IN\_YEAR;

averageYearTemperature.fahrenheit =

32 + (9.0 / 5) \* averageYearTemperature.celsius;

return averageYearTemperature;

}

unsigned int countBinaryZerosOrOnes(unsigned int N) {

if (N > 0 && N < 5740500) {

unsigned int tenthBit = (N >> 9) & 1;

unsigned int result = 0;

for (int i = 0; i < sizeof(N) \* 8; i++) {

result += ((N & 1) == tenthBit) ? 1 : 0;

N = N >> 1;

}

return result;

} else {

return -1;

}

}

*Лістинг проекту TestDriver:*

#include <iostream>

#include "ModulesHrebeniuk.h"

using namespace std;

void testFirstTask(int test\_case, int kWh, float expected) {

cout << "Тест-кейс №" << test\_case << endl;

cout << "Вхідні дані:\nkWh: " << kWh << endl;

float result = calculateElectricityConsume(kWh);

cout << "Результат: " << result << endl;

if (result == expected) {

cout << "Статус тест-кейса: passed" << endl;

} else {

cout << "Статус тест-кейса: failed" << endl;

}

cout << endl;

}

void testSecondTask(int test\_case, int averageMonthTemperature[MONTH\_IN\_YEAR], Temperature expected) {

cout << "Тест-кейс №" << test\_case << endl;

cout << "Вхідні дані:\naverageMonthTemperature: ";

for (int i = 0; i < MONTH\_IN\_YEAR; i++) {

cout << averageMonthTemperature[i];

if (i != (MONTH\_IN\_YEAR - 1)) cout << ", ";

}

cout << endl;

Temperature result = averageYearTemperature(averageMonthTemperature);

cout << "Результат:" << endl;

cout << "Celsius: " << result.celsius << endl;

cout << "Fahrenheit: " << result.fahrenheit << endl;

if (result.celsius == expected.celsius && result.fahrenheit == expected.fahrenheit) {

cout << "Статус тест-кейса: passed" << endl;

} else {

cout << "Статус тест-кейса: failed" << endl;

}

cout << endl;

}

void testThirdTask(int test\_case, unsigned int N, unsigned int expected) {

cout << "Тест-кейс №" << test\_case << endl;

cout << "Вхідні дані:\nN: " << N << endl;

unsigned int result = countBinaryZerosOrOnes(N);

cout << "Результат: " << result << endl;

if (result == expected) {

cout << "Статус тест-кейса: passed" << endl;

} else {

cout << "Статус тест-кейса: failed" << endl;

}

cout << endl;

}

int main()

{

system("chcp 65001 & cls");

// Task 9.1

cout << "=== TS\_9\_1 ===" << endl;

testFirstTask(1, 130, 17009.58984375);

testFirstTask(2, 151, 36533.6953125);

testFirstTask(3, 515, 124601.671875);

testFirstTask(4, -1, 0);

testFirstTask(5, 0, 0);

testFirstTask(6, 820, 437918.53125);

testFirstTask(7, 155, 37501.4765625);

testFirstTask(8, 80, 10467.4404296875);

testFirstTask(9, 1500, 801070.5);

testFirstTask(10, 1331, 710816.5625);

// Task 9.2

cout << "=== TS\_9\_2 ===" << endl;

int testCasesInput[10][MONTH\_IN\_YEAR] = {

{-6, 1, 20, -18, -7, 15, 25, -17, -28, 26, -30, 8}, // TC-01

{-23, 29, -1, 4, 2, -13, 15, 1, 15, -10, 18, -15}, // TC-02

{-6, 14, -5, -22, 12, -1, -30, -22, -29, 20, -10, -5}, // TC-03

{4, -14, 10, 8, -19, 10, -13, -12, 9, 19, 23, -20}, // TC-04

{-25, 9, 11, 22, -29, 0, -24, -3, -15, 3, 7, 30}, // TC-05

{-27, 7, -23, -25, -1, -3, 0, -28, 15, -21, 13, 27}, // TC-06

{22, 30, -16, 0, 18, -22, 10, 25, 20, -7, 16, 21}, // TC-07

{25, 24, -11, -20, -1, 26, -21, 2, 4, -12, 9, -28}, // TC-08

{17, -20, -25, 1, -8, 20, -3, -17, 19, 14, 13, 9}, // TC-09

{24, -6, 5, 13, 19, -9, 5, 14, -14, 24, 26, 17} // TC-10

};

Temperature testCasesOutput[10];

testCasesOutput[0].celsius = -0.91666668653488159;

testCasesOutput[0].fahrenheit = 30.350000381469727;

testSecondTask(1, testCasesInput[0], testCasesOutput[0]); // TC-01

testCasesOutput[1].celsius = 1.8333333730697632;

testCasesOutput[1].fahrenheit = 35.299999237060547;

testSecondTask(2, testCasesInput[1], testCasesOutput[1]); // TC-02

testCasesOutput[2].celsius = -7;

testCasesOutput[2].fahrenheit = 19.399999618530273;

testSecondTask(3, testCasesInput[2], testCasesOutput[2]); // TC-03

testCasesOutput[3].celsius = 0.4166666567325592;

testCasesOutput[3].fahrenheit = 32.75;

testSecondTask(4, testCasesInput[3], testCasesOutput[3]); // TC-04

testCasesOutput[4].celsius = -1.1666666269302368;

testCasesOutput[4].fahrenheit = 29.899999618530273;

testSecondTask(5, testCasesInput[4], testCasesOutput[4]); // TC-05

testCasesOutput[5].celsius = -5.5;

testCasesOutput[5].fahrenheit = 22.100000381469727;

testSecondTask(6, testCasesInput[5], testCasesOutput[5]); // TC-06

testCasesOutput[6].celsius = 9.75;

testCasesOutput[6].fahrenheit = 49.549999237060547;

testSecondTask(7, testCasesInput[6], testCasesOutput[6]); // TC-07

testCasesOutput[7].celsius = -0.25;

testCasesOutput[7].fahrenheit = 31.549999237060547;

testSecondTask(8, testCasesInput[7], testCasesOutput[7]); // TC-08

testCasesOutput[8].celsius = 1.6666666269302368;

testCasesOutput[8].fahrenheit = 35;

testSecondTask(9, testCasesInput[8], testCasesOutput[8]); // TC-09

testCasesOutput[9].celsius = 9.8333330154418945;

testCasesOutput[9].fahrenheit = 49.700000762939453;

testSecondTask(10, testCasesInput[9], testCasesOutput[9]); // TC-10

// Task 9.3

cout << "=== TS\_9\_3 ===" << endl;

testThirdTask(1, 2527920, 9);

testThirdTask(2, 2840999, 19);

testThirdTask(3, 5116662, 12);

testThirdTask(4, 1280412, 22);

testThirdTask(5, 1698360, 11);

testThirdTask(6, 413992, 25);

testThirdTask(7, 4027283, 14);

testThirdTask(8, 5401843, 19);

testThirdTask(9, 2933116, 21);

testThirdTask(10, 5156645, 13);

return 0;

}

**Задача 9.4**

Вхідні дані: task – символьний тип char – яку фукнцію потрібно виконати з модуля ModulesHrebeniuk (“n”, “v”, “c”, “x”)

Вихідні дані: результат виконання обраної функції

За допомогою конструкції switch-case визначити яку саме функцію потрібно викликати, потім запитати у користувача параметри для цієї функції та вивести результат виклику функції. Якщо користувач обрав неіснуючу функцію, видати звуковий сигнал про помилкове введення. Після виконання програми запропонувати користувачеві завершити виконання застосунку.

*Лістинг проекту Hrebeniuk\_task:*

#include <iostream>

#include <conio.h>

#include "Copyright.h"

#include "ModulesHrebeniuk.h"

using namespace std;

int main()

{

char task;

do {

cout << printCopyright() << endl;

cout << "Оберіть функцію, що бажаєте виконати (\"n\", \"v\", \"c\", \"x\"): ";

cin >> task;

cout << endl;

switch (task) {

case 'n': {

int x;

cout << "x: ";

cin >> x;

int y;

cout << "y: ";

cin >> y;

int z;

cout << "z: ";

cin >> z;

cout << "Результат функції s\_calculation(): " << s\_calculation(x, y, z) << endl;

break;

}

case 'v': {

int kWh;

cout << "kWh: ";

cin >> kWh;

cout << "Результат функції 9.1: " << calculateElectricityConsume(kWh) << endl;

break;

}

case 'c': {

int averageMonthTemperature[MONTH\_IN\_YEAR];

for (int i = 0; i < MONTH\_IN\_YEAR; i++) {

cout << "Введіть середню температуру за " << i + 1 << "-ий місяць: ";

cin >> averageMonthTemperature[i];

}

cout << endl;

Temperature result = averageYearTemperature(averageMonthTemperature);

cout << "Результат функції 9.2:" << "\nCelsius: " << result.celsius << "\nFahrenheit: " << result.fahrenheit << endl;

break;

}

case 'x': {

unsigned int N;

cout << "N: ";

cin >> N;

cout << "Результат функції 9.3: " << countBinaryZerosOrOnes(N) << endl;

break;

}

default: {

cout << "\aВи вказали неіснуючу функцію, спробуйте ще раз!" << endl;

break;

}

}

cout << "\nНатисніть \"z\" або \"Z\" для повторення виконання програми";

task = getch();

} while (task == 'z' || task == 'Z');

return 0;

}

**Висновок**

У цій лабораторній роботі я набув вмінь та практичних навичок реалізації технології модульного програмування, застосування арифметичних, логічних, побітових операцій та реалізації конструкцій розгалуження й повторення.

На початку лабораторної роботи було виконано аналіз вимог, проектування архітектури та проектування програмних модулів розв'язування задач 9.1, 9.2 та 9.3. При проектуванні потрібно було продумати алгоритм роботи кожної функції, правильно визначити прототип та декомпозувати задачу так, щоб визначити суть та обмеження кожної задачі. Одержаний результат було задокументовано й включено до звіту.

Наступною задачею було розроблення трьох тест-сьютів для виконання подальшого модульного тестування. При розробці цих тест-сьютів були виконані ручні обчислення, щоб визначити очікуваний результат роботи кожного тест-кейса. Були включені негативні тест-кейси, що тестують функції на помилкові вхідні дані. Отриманий результат був задокументований й включений до звіту.

Програмна реалізація нових задач була здійснена у теці лаборотної роботи №8, а саме в існуючому проекті ModulesHrebeniuk.

Під час реалізації задачі 9.1 була додана конструкція розгалуження, що визначає ціну за 1 кВт. Якщо за місяць спожито до 150 кВт**⋅**год включно, то за 1 кВт**⋅**год сплачується 130,843 коп. Якщо ж було спожито від 150 до 800 кВт**⋅**год включно, то 241,945 коп. Остання умова, якщо понад 800 кВт**⋅**год, то 534,047 коп. за 1 кВт**⋅**год.

Під час реалізації задачі 9.2 я використовував масив цілочисельного типу int, щоб на вхід отримати середній температурний показник кожного місяця за рік. Для визначення температури за шкалою Фаренгейта було використано формулу, що надається у завданні до варіанту. Результатом функції є дані складеного типу struct, а саме struct Temperature з параметрами: float celsius та float fahrenheit.

Для реалізації задачі 9.3 потрібно було працювати з бітами, побітовими операторами. В минулих лабораторних роботах вже зустрічалась робота з бітами, тому було простіше реалізувати це завдання. Була виконана перевірка на сумісність з умовою, а саме N > 0 та N < 5740500. Для визначення десятого біту потрібно було виконати зсув на 9 бітів та використання побітового «І» для отримання значення саме десятого біту. В подальшій реалізації використовувався тернарний оператор, що перевіряв біт на рівність з шуканим (змінна tenthBit). Функція повертає кількість двійкових нулів або одиниць, в залежності від значення десятого біту числа N.

Завершуючим етапом було доповнення заголовкового файлу прототипами нових функцій.

Для тестування тестових наборів було реалізовано тестовий драйвер під назвою проекту TestDriver. Було реалізовано протоколювання процесу тестування, а саме: виведення вхідних даних, отриманий від функції результат та статус кожного тест-кейса (passed/failed). Під час виконання роботи тестового драйверу усі тестові кейси кожного з трьох задач були завершені зі статусом passed.

Завдання 9.4 також вимагає попереднього аналізу і постановки задачі, виконання аналізу вимог до ПЗ, проектування архітектури та детальне проектування ПЗ для розв’язування цієї задачі. Суть задачи полягала в тому, щоб за введеним користувачем символом викликалась певна функція з модулю ModulesHrebeniuk. Кожна функція мала свій унікальний символ, а саме: “n” – s\_calculation(), “v” – функція задачі 9.1, “c” – функція задачі 9.2 та “x” – функція задачі 9.3. Якщо користувач введе інший символ, буде повідомлено про помилку та пролунає звуковий сигнал. Для повторення виконання програми виводиться запит на закінчення роботи застосунка. Якщо користувач хоче продовжити роботу, потрібно натиснути “z” або “Z”. Все це було задокументовано й включено до звіту. Також було розроблено тест-сьют для виконання системного тестування ПЗ, який складається з п’яти тест-кейсів. Після виконання системного тестування всі тест-кейси були завершені зі статусом passed.

Завдяки цій лабораторній роботі я покращив свої знання та практичні навички у проектуванні та розробці модульних проектів, розробці тестових драйверів, роботі з арифметичними, логічними та побітовими операціями. Мені сподобалось виконувати цю роботу, я знайшов для себе багато нової інформації та виконувати завдання було цікаво.

**Контрольні запитання і завдання**

**1.** *Яке призначення операцій інкремента та декремента? В чому полягає відмінність між їх префіксною і постфіксною формами у мові програмування С/С++?*

Операції інкремента (++) та декремента (--) використовуються для збільшення або зменшення значення змінної на одиницю. Префіксна форма (++i або --i) спочатку змінює значення змінної, а потім повертає його, тоді як постфіксна форма (i++ або i--) спочатку повертає поточне значення змінної, а потім змінює його.

**2.** *Яке значення матиме mch при char mch = 'D'; --mch; ? Поясніть отриманий результат.*

Значення mch буде 'C', оскільки --mch зменшує значення mch на одиницю. В таблиці ASCII 'D' має код 68, а 'C' - 67.

**3.** *Якими є призначення й синтаксис запису складових операторів присвоювання в мові програмування С/С++?*

Оператори складового присвоювання включають +=, -=, \*=, /=, %= тощо. Вони використовуються для зміни значення змінної шляхом виконання операції та присвоєння результату тій самій змінній.

**4.** *Перелічіть оператори логічних операцій С/С++, типи їх операндів і результату.*

Оператори логічних операцій включають && (логічне І), || (логічне АБО), ! (логічне НЕ). Типи операндів - bool, а результат також bool.

**5.** *Яким буде значення comp при bool comp = (!0 && 1) == 0; ? Отриманий результат обґрунтуйте.*

Значення comp буде false, оскільки вираз (!0 && 1) є істинним, але порівнюється з 0 (false), тому весь вираз (!0 && 1) == 0 є неправдивим (false).

**6.** *Перелічіть оператори порозрядних операцій С/С++, типи їх операндів і результату виконання.*

Оператори порозрядних операцій включають:

- & (порозрядне І)

- | (порозрядне АБО)

- ^ (порозрядне виключне АБО)

- ~ (порозрядне НЕ)

- << (зсув вліво)

- >> (зсув вправо)

Типи операндів для цих операторів зазвичай є цілочисельними типами, а результатом є значення того ж типу, що й операнди.

**7.** *Розташуйте в порядку спадання пріоритету виконання наступних операцій: логічні, арифметичні, порівняння, інкремент/декремент, складові операції присвоювання.*

Порядок пріоритету виконання операцій:

- Інкремент/декремент (++, --)

- Арифметичні (\*, /, %, +, -)

- Порівняння (==, !=, <, >, <=, >=)

- Логічні (&&, ||)

- Складові операції присвоювання (=, +=, -=, \*=, /=, %=)

**8.** *Що розуміють під асоціативністю операторів мови програмування? Наведіть приклад.*

Асоціативність операторів визначає порядок, в якому оператори з однаковим пріоритетом групуються при відсутності дужок. Наприклад, вираз a - b - c за замовчуванням обчислюється як (a - b) - c через лівоасоціативність оператора -.

**9.** *Яке значення матиме n при: int n = 0; n = !n<<1&1; ?*

Для виразу int n = 0; n = !n<<1&1; значення n буде 0.

**10.** *Якими є призначення і синтаксис запису операторів повного й неповного розгалуження С/С++?*

// Повне розгалуження

if (умова) {

// інструкції, якщо умова істинна

} else {

// інструкції, якщо умова хибна

}

// Неповне розгалуження

if (умова) {

// інструкції, якщо умова істинна

}

**11.** *Сформулюйте алгоритм виконання тернарного оператора (операції умови) мови програмування С/С++.*

Тернарний оператор у С/С++ використовується для вибору одного з двох значень залежно від умови. Його алгоритм: умова ? вираз1 : вираз2;

Якщо умова істинна, то виконується вираз1, інакше - вираз2.

**12.** *У чому полягає відмінність між результатом виконання оператора break та оператора continue? Перелічіть випадки, у яких є доцільним використання кожного з них під час реалізації програмного забезпечення.*

Оператор break припиняє виконання циклу або оператора switch, виходячи з нього. Оператор continue припиняє поточну ітерацію циклу і переходить до наступної ітерації. break доцільно використовувати для виходу з циклу, коли мета досягнута або коли потрібно негайно припинити виконання циклу. continue використовується для пропуску певних умов всередині циклу без виходу з нього.

**13.** *Яке призначення циклів у програмуванні та які розрізняють їх види у мові програмування С/С++?*

Цикли в програмуванні використовуються для повторення блоку коду декілька разів. У С++ існують такі види циклів:

- for

Використовується, коли відомо точна кількість ітерацій.

- while

Виконується, поки умова істинна.

- do…while

Виконується хоча б один раз, а потім продовжується, поки умова істинна.

**14.** *Сформулюйте алгоритм виконання оператора switch. Якого типу мають бути його перемикаючий і константні вирази?*

Оператор switch використовується для виконання різних дій залежно від різних умов. Алгоритм:

1. Вираз у операторі switch обчислюється.

2. Значення цього виразу порівнюється з кожним випадком (case) в операторі switch.

3. Якщо значення виразу співпадає зі значенням якого-небудь випадку, виконуються інструкції відповідного випадку.

4. Якщо жоден випадок не збігається зі значенням виразу, виконуються інструкції випадку за замовчуванням (якщо такий випадок є).

5. Виконання оператора switch завершується.

Перемикаючий вираз може бути будь-яким виразом, що повертає значення. Це може бути числове значення (ціле число, дійсне число), символьне значення або навіть рядок. Константні вирази повинні бути константами. Це можуть бути цілі числа, символи, перечислення, або константні вирази.

**15.** *Яким є синтаксис запису циклів for, while, do…while С/С++? Сформулюйте алгоритми їх виконання.*

// for

for (ініціалізація; умова; інкремент) {

// блок коду

}

// while

while (умова) {

// блок коду

}

// do…while

do {

// блок коду

} while (умова);

Алгоритми виконання:

- for ініціалізує змінну, перевіряє умову; якщо вона істинна, виконує блок коду, а потім інкрементує змінну.

- while перевіряє умову перед кожною ітерацією; якщо вона істинна, виконує блок коду.

- do…while виконує блок коду один раз, а потім перевіряє умову; якщо вона істинна, продовжує виконання.

**16.** *Яке призначення кваліфікатора типів const С/С++? Наведіть приклад і перелічіть випадки доцільності його застосування.*

Кваліфікатор типу const призначений для позначення, що значення змінної не може бути змінено після її ініціалізації.

Наприклад:

const int x = 5;

x = 10; // Помилка: x є константою і не може бути змінено.

Випадки застосування:

1. Захист від ненавмисних змін значень змінних.

2. Оптимізація коду компілятором.

3. Забезпечення безпеки у великих проектах, де важлива стабільність даних.

**17.** *Який з виразів оператора for (ініціалізації, умови, модифікації/ оновлення) є обов’язковим у С/С++? Відповідь обґрунтуйте*

У операторі for у C/C++ обов’язковим є лише вираз умови. Ініціалізація та модифікація/оновлення не є обов’язковими і можуть бути виключені, якщо це необхідно.

**18.** *Що буде виведено об’єктом cout при: short int b = 0; cout << hex << ( ~b&0x80 ); ?*

Вираз cout << hex << ( ~b&0x80 ); виведе 80.

**19.** *Сформулюйте алгоритм перетворення функції С/С++ у функцію статичної бібліотеки С/С++, а також спосіб її використання під час реалізації програмного забезпечення.*

Для перетворення функції C/C++ у функцію статичної бібліотеки, потрібно:

1. Створити статичну бібліотеку

2. Додати код функції у вихідний код бібліотеки

3. Додати прототип функції у окремий заголовковий файл

4. Скомпілювати статичну бібліотеку у об’єктний файл

5. Підключити статичну бібліотеку до програми за допомогою лінкера

6. Підключити заголовковий файл статичної бібліотеки до програми

**20.** *Наведіть приклад опису константи у заголовковому файлі С/С++ та її використання у програмі.*

// example.h

#ifndef EXAMPLE\_H

#define EXAMPLE\_H

const int MONTH\_IN\_YEAR = 12;

#endif // EXAMPLE\_H

// main.cpp

#include "example.h"

int main() {

int array[MONTH\_IN\_YEAR];

// Використання константи MONTH\_IN\_YEAR

return 0;

}