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”

На тему:

“ Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів”

ВИКОНАВ

студент академічної групи КН-22

Бездольний К. О.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Собінов О. Г.

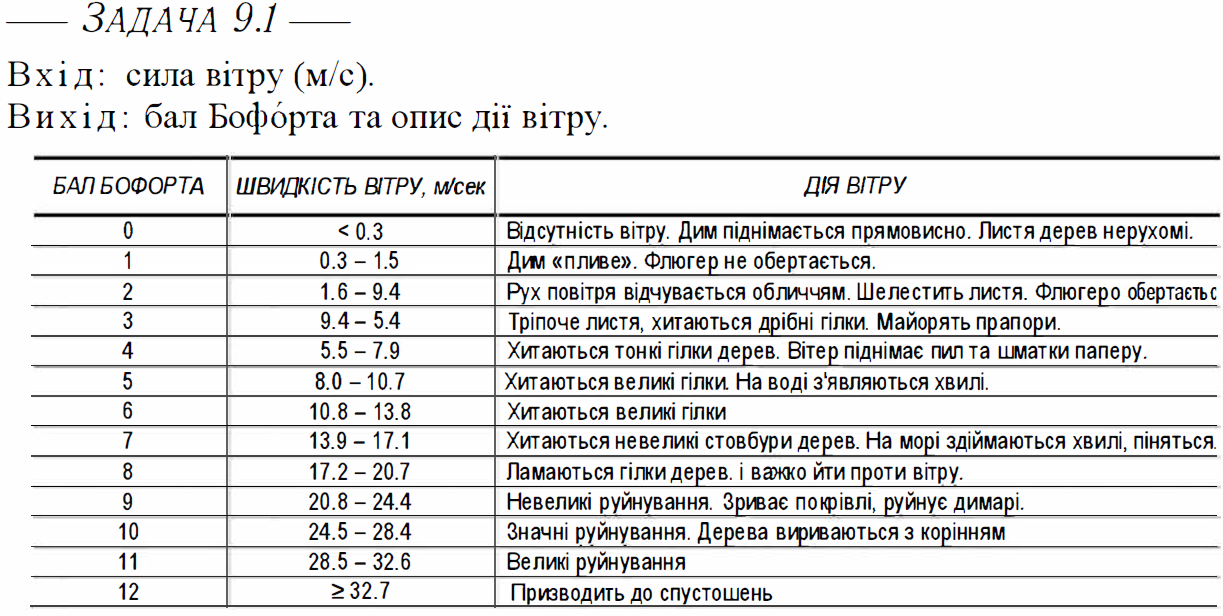
м. Кропивницький 2023

*Мета роботи :* полягає у набутті ґрунтовних вмінь і практичнихнавичок реалізації технології модульного програмування,застосування операторів С/С++ арифметичних, логічних, побітовихоперацій, умови, циклів та вибору під час розроблення статичнихбібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

*Завдання :*

1. Реалізувати функції розв’язування задач 9.1–9.3 як складовихстатичної 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 13**

*Аналіз умови до задачі 9.1.* Шкала Бофорта має 13 ступенів (від 0 до 12), і зазвичай використовується для опису сили вітру на морі, але також може бути застосована на суходолі. Користувач вводить силу вітру та за цим параметром знаходиться відповідний діапазон швидкості в шкалі Бофорта. Після цього, повертається опис дії вітру на довкілля та його потенційний вплив на людей та інфраструктуру.

*Лістниг програмного коду:*

string bofort(float wind) {

if (wind<0.3) return "Бал Бофорта: 0. Вiдсутнiсть вiтру. Дим пiднiмаеться прямовисно. Листя дерев нерухомi.";

else if (wind<1.5) return "Бал Бофорта: 1. Дим «пливе». Флюгер не обертаеться.";

else if (wind<3.4) return "Бал Бофорта: 2. Рух повiтря вiдчуваеться обличчям. Шелестить листя. Флюгеро обертається спокійно.";

else if (wind<5.4) return "Бал Бофорта: 3. Трiпоче листя, хитаються дрiбнi гiлки. Майорять прапори.";

else if (wind<7.9) return "Бал Бофорта: 4. Хитаються тонкi гiлки де рев. Вiтер пiднiмае пил та шматки паперу.";

else if (wind<10.7) return "Бал Бофорта: 5. Хитаються великi гiлки. На водi з'являються хвилi.";

else if (wind<13.8) return "Бал Бофорта: 6. Хитаються великi гiлки";

else if (wind<17.1) return "Бал Бофорта: 7. Хитаються невеликi стовбури де рев. На морi здiймаються хвилi, пiняться.";

else if (wind<20.7) return "Бал Бофорта: 8. Ламаються гiлки дере в. i важко йти проти вiтру.";

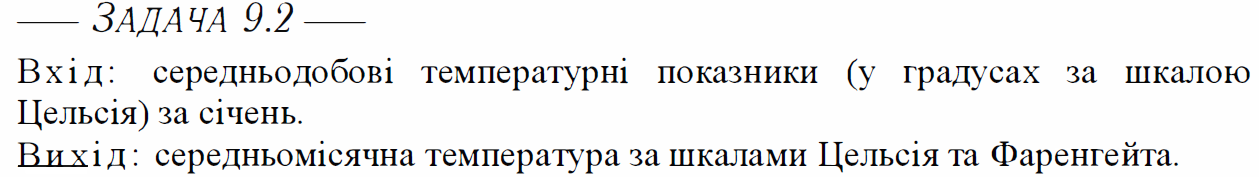
else if (wind<24.4) return "Бал Бофорта: 9. Невеликi руйнування. Зривае покрiвлi, руйнуе димарi.";

else if (wind<28.4) return "Бал Бофорта: 10. Значнi руйнування. Дерева вириваються з корiнням";

else if (wind<32.6) return "Бал Бофорта: 11. Великi руйнування";

else return "Бал Бофорта: 12. Призводить до спустошень";

}



*Аналіз задачі 9.2* Користувач вводить середньодобові температурні показники за січень, після обчислюється середньомісячна температура у градусах Цельсія та перетворюється на градуси Фаренгейта.

*Лістниг програмного коду:*

double celsius\_to\_fahrenheit(double temp\_C) {

return (temp\_C \* 9 / 5) + 32;

}

double average\_temperature\_C(const double temp\_C[], int days) {

double sum = 0;

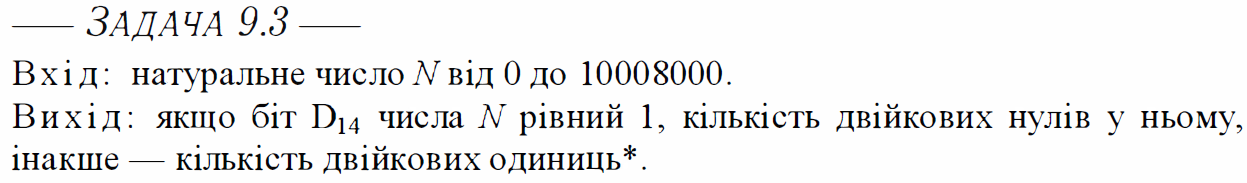
for (int i = 0; i < days; i++) {

sum += temp\_C[i];

}

return sum / days;

}



*Аналіз задачі 9.3* Користувач вводить число від 0 до 10008000. Спочатку число потрібно перетворити в двійкову систему числення. Потім після цього порівняти 14 біт, якщо він дорівнює 0, то вивести кількість двійкових одиниць, інакше – кількість двійкових нулів.

*Лістниг програмного коду:*

string binary\_count(unsigned int iNumber) {

const unsigned int maxNumber = 10008000;

stringstream resultStr;

if(iNumber > maxNumber) {

resultStr << "Your number is more than " << maxNumber << " or less than zero!";

return resultStr.str();

}

bitset<32> bNumber(iNumber);

size\_t ones = bNumber.count();

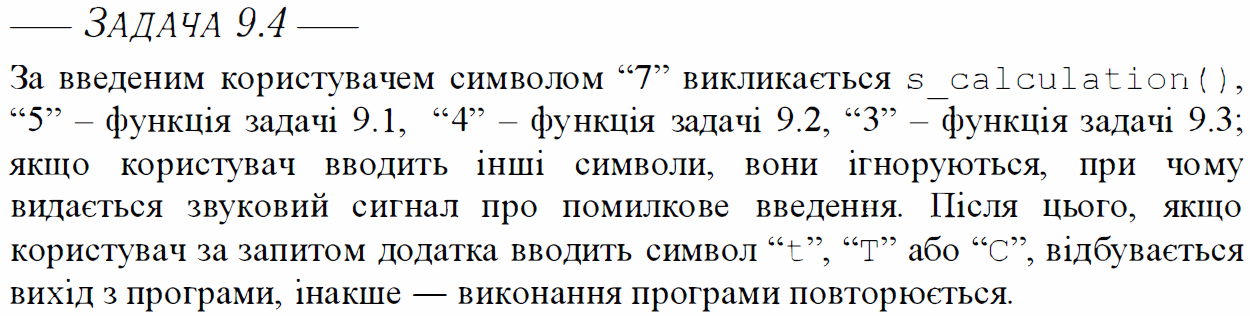
size\_t zeros = bNumber.size() - ones;

if(bNumber.test(13)) resultStr << "Zeros count: " << zeros;

else resultStr << "Ones count: " << ones;

return resultStr.str();

}



*Аналіз задачі 9.4* Користувач вводить символ с клавіатури, якщо цей символ “7” викликається s\_calculation (), “5” – функція задачі 9.1, “4” – функція задачі 9.2, “3” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “t”, “T” або “c”, відбувається вихід з програми, інакше виконання програми повторюється.

*Лістниг програмного коду:*

#include <iostream>

#include <windows.h>

#include "ModulesBezdolny.h"

using namespace std

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char input;

bool exit = false;

while (!exit) {

cout << endl << "Введіть символ ('t', 'T' або 'С' для виходу з програми): ";

cin >> input;

input = tolower(input);

switch (input) {

case '7':

{

int x, y, z;

cout << "Введiть x, y, z: ";

cin >> x >> y >> z;

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

break;

}

case '5':

{

float wind\_power;

cout << "Enter wind power: ";

cin >> wind\_power;

cout << bofort(wind\_power) << endl;

break;

}

case '4':

{

int days;

cout << "Enter days count: ";

cin >> days;

double temp\_C[days];

cout << "Enter temperature for every day(" << days << "): ";

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

cin >> temp\_C[i];

}

double average\_temp\_C = average\_temperature\_C(temp\_C, days);

cout << "Середньомісячна температура у градусах Цельсія: " << average\_temp\_C << endl;

cout << "Середньомісячна температура у градусах Фаренгейта: " << celsius\_to\_fahrenheit(average\_temp\_C) << endl;

break;

}

case '3':

{

unsigned int num;

cout << "Enter num: ";

cin >> num;

cout << binary\_count(num) << endl;

break;

}

case 't':

exit = true;

break;

case 'c':

exit = true;

break;

default:

cout << '\a';

}

}

return 0;

}

**Висновок**

Одним із спостережень виконання лабораторної роботи є факт вирішення задач із попередніх лабораторних, проте з використанням модульної парадигми та більш ширшого спектру взаємодії з середовищем розробки. Керуючись строгою постановкою задачі, попередньо було виявлено та ліквідовано можливі проблеми:

* Слушною думкою було попереднє визначення кодування файлу вихідного коду (UTF-8, адже с цим кодуванням, на відміну від CP-1251 та CP-866 не виникає проблем з відображенням файлу коду на сторінці GitHub);
* Також, передбачено критичне завершення програми (або - виведення результату в консоль) у випадку отримання не коректних вхідних даних. Так в окремих модулях, хибні значення визначаються 0 (завдання 9.1: нулем визначається час роботи), або ж як -1 (в інших завданнях, окрім 9.4), що одразу свідчить про порушення умов алгоритмізації та визначення результату TestCase як Failed.

Виходячи з технічних проблем, підключення стандартних динамічних бібліотек компілятора MinGW через налаштування проекту Code::Blocks було опущено. Необхідні для роботи застосунків бібліотеки переміщені в директорію Software.

Для завдання 9.1 прийняті декілька рішень, щодо оформлення програмного коду:

По-перше - створено окрему структуру, котра зберігає суму З/П на різних етапах та суми всіх податків, залежно від З/П. Тип повернення функції є типом даної структури. Прибігання до цього викликане тим, що при майбутньому тестуванні модуля TestDriver матиме доступ одразу до декількох закритих змінних структури.

По-друге, для випадку, коли в аргументі-масиві присутній елемент з від’ємним значенням, його буде переініціалізовано нулем заради запобігання не коректних результатів виконання програми. В консоль виводиться сповіщення щодо цього випадку.

Аналогічно до задачі 9.1, функція для завдання 9.2 має повертати тип типу структури, котра є завчасно створеною та зберігає в собі всі (дані в умові) відповідники розмірів. Також, якщо отриманий розмір (в якості аргументу) не відноситься до жодного із зазначених до української системи розмірів - функція повертає структуру, кожне значення якої визначене як -1 (що автоматично визначає результат TestCase як Failed). Проектне рішення, щодо оптимізації коду для даної задачі: за таблицею відповідників розмірів була визначена закономірність і з метою запобігання надлишкових рядків коду, змінні міжнародних відповідників були ініцалізовані формулами залежності від українського розміру.

Передбачено отримання хибних вхідних даних і для функції задачі 9.3: повернення в результат -1, що автоматично визначає хибний результат TestCase. Згідно з того, що зчитування бітів за стандартом відбувається з кінця числа - число було попередньо інвертоване. Надалі реалізовано фрагмент коду, що безпосередньо відповідає за побітові маніпуляції:

Всередині циклу, ітерованим 32 рази (дане рішення викликане тим, що число типу int займає саме 32 біти) зберігаємо останнє входження біта 1. Це необхідно для подальшого відсікання “зайвих” нулів з лічильника їх кількості. Після цього, до змінної кількості бітів додається результат логічного порівняння рівності поточного біту з бітом, заданим в умові задачі (інакше кажучи, якщо істина - то до лічильника додається 1, якщо хиба - 0 або нічого). Виведення результату реалізовано за допомогою оператору множинного вибору switch, що перевіряє значення N біту.

Так як для завдання 9.4 попередньо необхідно мати відлагоджений код проекту статичної бібліотеки та згідно з того що кожне вхідне значення відноситься до числового типу даних, в даному завданні на вхід аргументам функцій надсилається випадково згенероване значення. Для зручності використання та для коректної реалізації виходу із програми, виконано наступну маніпуляцію: оператор множинного вибору switch, що приймає символ введений з клавіатури, поміщений у вічний цикл. Для виходу з циклу необхідно ввести символ, що зазначено в умові завдання. Вихід буде виконано не стандартним завершенням циклу, а напряму буде завершено роботу застосунку.

Загалом: лабораторна робота дозволила повноцінно закріпити навички використання статичних бібліотек та врахування всіх аспектів пов’язаних з ними. Паралельно, покращене розуміння маніпулювання бітами певного числа та роботу з функціями що в якості аргументів і/або результату повернення використовують змінні складених типів (структури, масиви, і. т. д.). Врахований факт необхідності кодування файлу, що надає можливість до коректного перегляду його вмісту на сторінці ресурсу GitHub.

**Модульний тест до задачі 9.1**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_1 |
| Назва проекта / ПЗ  Name of Project / Software | bofort() |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Бездольний Кирило |
| Виконавець  Implementer | Бездольний Кирило |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TCM\_1 | Вхід: 14.1 | Вихід: Бал Бофорта: 7. Хитаються невеликi стовбури де рев. На морi здiймаються хвилi, пiняться. | PASSED |
| TCM\_2 | Вхід: 0.6 | Вихід: Бал Бофорта: 1. Дим «пливе». Флюгер не обертаеться. | PASSED |
| TCM\_3 | Вхід: 0.0 | Вихід: Бал Бофорта: 0. Вiдсутнiсть вiтру. Дим пiднiмаеться прямовисно. Листя дерев нерухомi. | PASSED |
| TCM\_4 | Вхід: -1.0 | Вихід: Бал Бофорта: 0. Вiдсутнiсть вiтру. Дим пiднiмаеться прямовисно. Листя дерев нерухомi. | PASSED |
| TCM\_5 | Вхід: 100.0 | Вихід: Бал Бофорта: 12. Призводить до спустошень | PASSED |

**Модульний тест до задачі 9.2**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_2 |
| Назва проекта / ПЗ  Name of Project / Software | average\_temperature\_C() |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Бездольний Кирило |
| Виконавець  Implementer | Бездольний Кирило |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TCM\_1 | Вхід: -5.0, -4.5, -4.0, -3.5, -3.0, -2.5, -2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0 | Вихід: 2.5, 36.5 | PASSED |
| TCM\_2 | Вхід: -6.5, -6.0, -5.5, -5.0, -4.5, -4.0, -3.5, -3.0, -2.5, -2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5 | Вихід: 1, 33.8 | PASSED |
| TCM\_3 | Вхід: -10.0, -9.5, -9.0, -8.5, -8.0, -7.5, -7.0, -6.5, -6.0, -5.5, -5.0, -4.5, -4.0, -3.5, -3.0, -2.5, -2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0 | Вихід: -2.5, 27.5 | PASSED |
| TCM\_4 | Вхід: 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 | Вихід: 0, 32 | PASSED |
| TCM\_5 | Вхід: 5.5, 5.0, 4.5, 4.0, 3.5, 3.0, 2.5, 2.0, 1.5, 1.0, 0.5, 0.0, -0.5, -1.0, -1.5, -2.0, -2.5, -3.0, -3.5, -4.0, -4.5, -5.0, -5.5, -6.0, -6.5, -7.0, -7.5, -8.0, -8.5, -9.0, -9.5 | Вихід: -2, 28.4 | PASSED |

**Модульний тест до задачі 9.3**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_3 |
| Назва проекта / ПЗ  Name of Project / Software | binary\_сount() |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Бездольний Кирило |
| Виконавець  Implementer | Бездольний Кирило |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TCM\_1 | Вхід: 655304 | Вихід: Zeros count: 19 | PASSED |
| TCM\_2 | Вхід:500403 | Вихід: Zeros count: 21 | PASSED |
| TCM\_3 | Вхід: 40378 | Вихід: Ones count: 10 | PASSED |
| TCM\_4 | Вхід: 1688 | Вихід: Ones count: 5 | PASSED |
| TCM\_5 | Вхід: 3264 | Вихід: Ones count: 4 | PASSED |

**Системний тест до задачі 9.4**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_4 |
| Назва проекта / ПЗ  Name of Project / Software | Bezdolny\_task |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Бездольний Кирило |
| Виконавець  Implementer | Бездольний Кирило |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TS-1 | 1. Ввести “7” 2. Ввести 12 3. Ввести 30 4. Ввести 5 5. Ввести “t” | Enter action  ==>7  Enter x  ==>12  Enter y  ==>30  Enter z  ==>5  Out  ==>-325.378  Enter action  ==>t | passed |
| TS-2 | 1. Ввести “5” 2. Ввести 13 3. Ввести “T” | Enter action  ==>5  Enter wind power  ==>13  Out  ==> Бал Бофорта: 6. Хитаються великi гiлки  Enter action  ==>t | passed |
| TS-3 | 1. Ввести “4” 2. Ввести “10” 3. Ввести -8 9 7 -6 0 -4 -7 -1 -2 -10 4. Ввести “c” | Enter action  ==>4  Enter days count  ==>10  Enter temperature for every day(10)  ==> -8 9 7 -6 0 -4 -7 -1 -2 -10  Out: ==> Середньомісячна температура у градусах Цельсія: -2.2  ==> Середньомісячна температура у градусах Фаренгейта: 28.04  Enter action  ==>c | passed |
| TS-4 | 1. Ввести “3” 2. Ввести 4568 3. Ввести “t” | Enter action  ==>3  Enter num  ==>4568  Out  ==>Ones count: 6  Enter action  ==>t | passed |
| TS-5 | 1. Ввести “2” 2. Ввести “j” 3. Ввести “c” | Enter action  ==>2  \*Звуковий сигнал\*  Enter action  ==>j  \*Звуковий сигнал\*  Enter action  ==>c | passed |