МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 4

з навчальної дисципліни

“Базові методології та технології програмування”

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

ВИКОНАВ

студент академічної групи КБ-23

Сергій Козирь

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Ганна ДРЄЄВА

Кропивницький – 2024

***Мета 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.

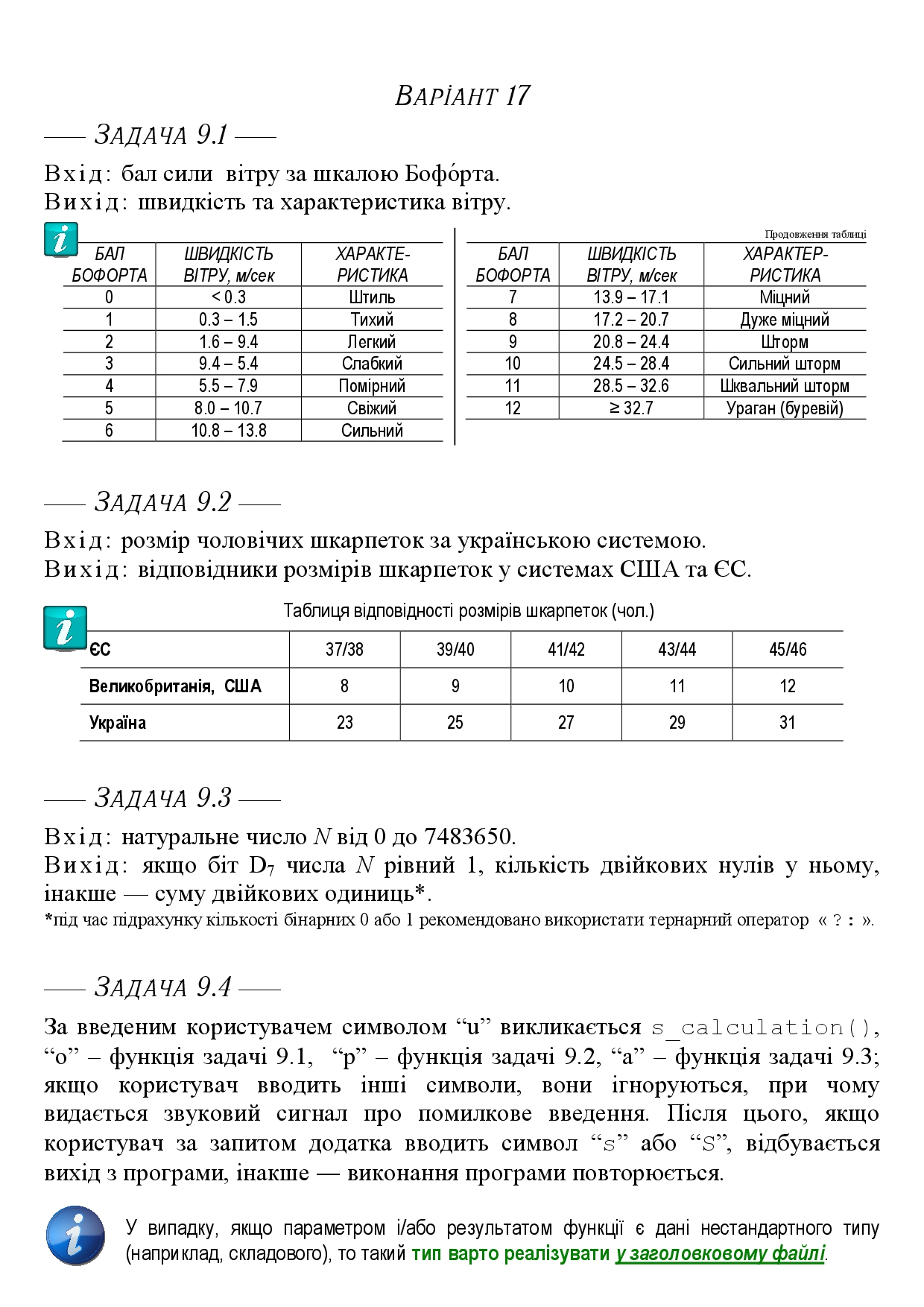
**Час виконання лабораторної роботи**

4 академічні години.

**Обладнання, матеріали, програмні засоби**

Для виконання лабораторної роботи необхідні: – персональний комп’ютер з ОС Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit або Mac OS X; – вільне кросплатформове середовище розроблення програмного забезпечення Code::Blocks (www.codeblocks.org) для відповідної платформи: Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit, або Mac OS X; – текстовий редактор (OpenOffice Writer, Microsoft Word або ін.); – файл-шаблон тестового набору Artifact\_TEST\_SUITE\_lab.doc; – Git-репозиторій <https://github.com/odorenskyi/student-name>.

**Завдання до лабораторної роботи**

****

**Завдання 9.1**

**Алгоритм**

Початок

1.Ввести бал Бофорта(від 0 до 12).

2.Вивід швидкість та характеристика вітру.

Кінець

**Лістинг**

#include <iostream>

#include <string>

using namespace std;

// Функція для визначення характеристики вітру

string determineWindCharacteristics(int beaufortScale) {

switch(beaufortScale) {

case 0:

return "Штиль";

case 1:

return "Тихий";

case 2:

return "Легкий";

case 3:

return "Слабкий";

case 4:

return "Помiрний";

case 5:

return "Свiжий";

case 6:

return "Сильний";

case 7:

return "Мiцний";

case 8:

return "Дуже мiцний";

case 9:

return "Шторм";

case 10:

return "Сильний шторм";

case 11:

return "Шквальний шторм";

case 12:

return "Ураган (буревiй)";

default:

return "Невiдомий";

}

}

// Функція для визначення швидкості вітру за шкалою Бофорта

double determineWindSpeed(int beaufortScale) {

switch(beaufortScale) {

case 0:

return 0.3;

case 1:

return 1.5;

case 2:

return 3.4;

case 3:

return 5.4;

case 4:

return 7.9;

case 5:

return 10.7;

case 6:

return 13.7;

case 7:

return 17.1;

case 8:

return 20.7;

case 9:

return 24.4;

case 10:

return 28.4;

case 11:

return 32.6;

case 12:

return 32.7; // Для випадку буревію

default:

return -1; // Повертаємо -1, якщо значення не відоме

}

}

int main() {

setlocale(LC\_ALL, "ukr");

int beaufortScale;

cout << "Введiть бал Бофорта (вiд 0 до 12): ";

cin >> beaufortScale;

double windSpeed = determineWindSpeed(beaufortScale);

string windCharacteristics = determineWindCharacteristics(beaufortScale);

if (windSpeed != -1) {

cout << "Швидкiсть руху: " << windSpeed << " м/с" << endl;

cout << "Характеристика: " << windCharacteristics << endl;

} else {

cout << "Невiдомий бал Бофорта." << endl;

}

return 0;

}

**Завдання 9.2**

**Алгоритм**

Початок

1.Введення значення чоловічих шкарпеток за українською системою.

2.Зрівнення цих значеннь зі значенням чоловічих шкарпеток в США та ЄС.

3.Вивід результатів.

Кінець

**Лістинг**

#include <iostream>

#include <map>

using namespace std;

// Функція для конвертації розміру шкарпеток з української системи у розмір у США

int convertToUS(int size) {

map<int, int> ukrToUS = {

{23, 8},

{25, 9},

{27, 10},

{29, 11},

{31, 12}

};

if (ukrToUS.find(size) != ukrToUS.end()) {

return ukrToUS[size];

} else {

return -1; // Повертаємо -1, якщо розмір не знайдено

}

}

// Функція для конвертації розміру шкарпеток з української системи у розмір у ЄС

string convertToEU(int size) {

map<int, string> ukrToEU = {

{23, "37/38"},

{25, "39/40"},

{27, "41/42"},

{29, "43/44"},

{31, "45/46"}

};

if (ukrToEU.find(size) != ukrToEU.end()) {

return ukrToEU[size];

} else {

return "Невiдомий"; // Повертаємо "Невідомий", якщо розмір не знайдено

}

}

int main() {

setlocale (LC\_ALL, "ukr");

int ukrSize;

cout << "Введiть розмiр чоловiчих шкарпеток за українською системою (23, 25, 27, 29, 31): ";

cin >> ukrSize;

int usSize = convertToUS(ukrSize);

string euSize = convertToEU(ukrSize);

if (usSize != -1) {

cout << "Вiдповiдний розмiр у США: " << usSize << endl;

} else {

cout << "Розмiр у США: Невiдомий" << endl;

}

cout << "Вiдповiдний розмiр у Європi: " << euSize << endl;

return 0;

}

**Завдання 9.3**

**Алгоритм**

Початок

1.Введення: натуральне число N

2.Створюємо обмеження для максимального числа.

3. якщо біт D3 числа N рівний 1, кількість двійкових нулів у ньому, інакше — кількість двійкових одиниць.

Кінець

**Лістинг**

#include <iostream>

#include <bitset>

using namespace std;

// Функція для підрахунку кількості двійкових нулів у числі

int countZeroes(int num) {

bitset<32> binary(num);

int count = 0;

for (int i = 0; i < 32; ++i) {

if (binary[i] == 0) {

count++;

}

}

return count;

}

// Функція для підрахунку суми двійкових одиниць у числі

int sumOnes(int num) {

bitset<32> binary(num);

int sum = 0;

for (int i = 0; i < 32; ++i) {

if (binary[i] == 1) {

sum++;

}

}

return sum;

}

int main() {

setlocale(LC\_ALL, "ukr");

int N;

cout << "Введiть натуральне число N (вiд 0 до 7483650): ";

cin >> N;

// Перевірка, чи введене число не перевищує 7483650

if (N > 7483650) {

cout << "Введене число перевищує максимально допустиме значення (7483650)." << endl;

return 1; // Повертаємо 1, щоб позначити помилку

}

// Перевірка біту D7

if (N & (1 << 7)) {

// Якщо біт D7 рівний 1

int zeroesCount = countZeroes(N);

cout << "Кiлькiсть двiйкових нулiв: " << zeroesCount << endl;

} else {

// Якщо біт D7 рівний 0

int onesSum = sumOnes(N);

cout << "Сума двiйкових одиниць: " << onesSum << endl;

}

return 0;

}

**Завдання 9.4**

**Алгоритм**

Початок

1.Вибір значення (u,o,p,a).

2.Вивід

Кінець

**Лістинг**

#include <iostream>

#include <cstdlib> // Для використання функції system("pause")

#include <Windows.h> // Для використання функції Beep

using namespace std;

// Функція для обробки випадку "u"

void s\_calculation() {

cout << "Виклик функції s\_calculation()" << endl;

// Ваш код для обробки випадку "u" тут

}

// Функція для обробки випадку "о"

void task\_9\_1() {

cout << "Виклик функції задачі 9.1" << endl;

// Ваш код для обробки випадку "о" тут

}

// Функція для обробки випадку "p"

void task\_9\_2() {

cout << "Виклик функції задачі 9.2" << endl;

// Ваш код для обробки випадку "p" тут

}

// Функція для обробки випадку "a"

void task\_9\_3() {

cout << "Виклик функції задачі 9.3" << endl;

// Ваш код для обробки випадку "a" тут

}

int main() {

char choice;

do {

cout << "Введіть символ ('u', 'o', 'p', 'a') або 's' для завершення: ";

cin >> choice;

switch (choice) {

case 'u':

s\_calculation();

break;

case 'o':

task\_9\_1();

break;

case 'p':

task\_9\_2();

break;

case 'a':

task\_9\_3();

break;

case 's':

case 'S':

return 0; // Вихід з програми

default:

// Якщо введений символ не відповідає жодному з варіантів, відтворюємо звуковий сигнал помилки

Beep(1000, 500); // Перший параметр - частота звуку (в герцах), другий - тривалість (в мілісекундах)

cout << "Помилкове введення. Спробуйте ще раз." << endl;

}

} while (true);

//system("pause"); // Використовуйте цей рядок, якщо ви використовуєте Windows і хочете зупинити консоль перед завершенням

return 0;

}

**Висновок**: В ході виконання лабораторної роботи я більше попрактикувався з використанням функцій і створення бібліотеки модулів, що дозволить ефективніше вирішувати поставлені задачі для створення програми. Бібліотеки модулів дають змогу винести функції, які використовує програма у окремий файл, таким чином різні люди можуть додавати елементи програми, як функції.

Також я 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.