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

ВИКОНАВ

студент академічної групи \_\_\_\_\_

\_\_\_\_\_\_\_\_\_ Барамба А.А

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ Павло УСІК

Кропивницький – 2022

**ЛАБОРАТОРНА РОБОТА №9**

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

**МЕТА:** набути 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.

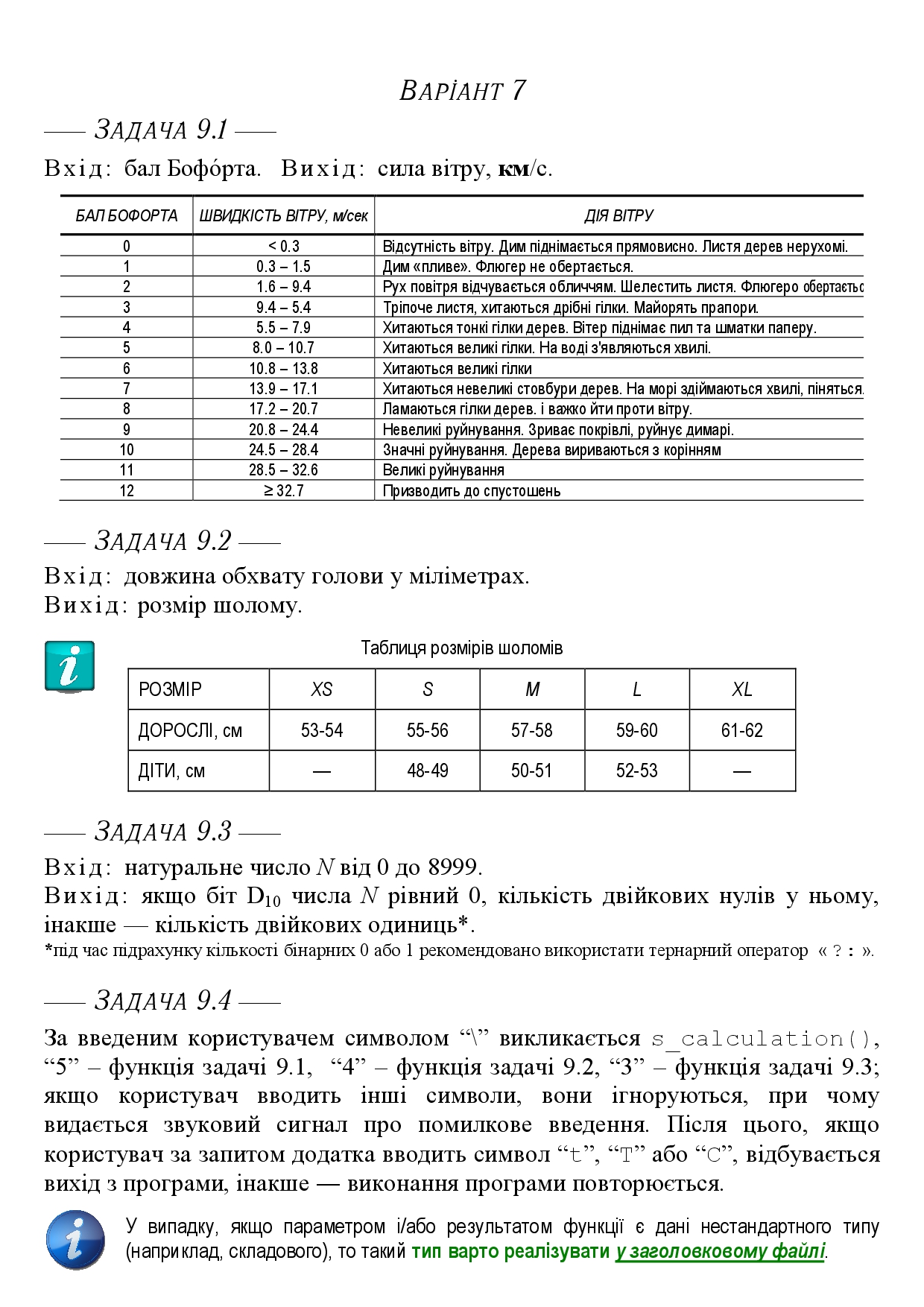
**ВАРІАНТ 7**

**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

**ХІД РОБОТИ**

Рисунок 1.1 – Умова задачі 9.1



**Строга постановка задачі**

Вхідні дані:

* Бал за шкалою Бофорта

Вихідні дані:

* Сила вітру у км/с

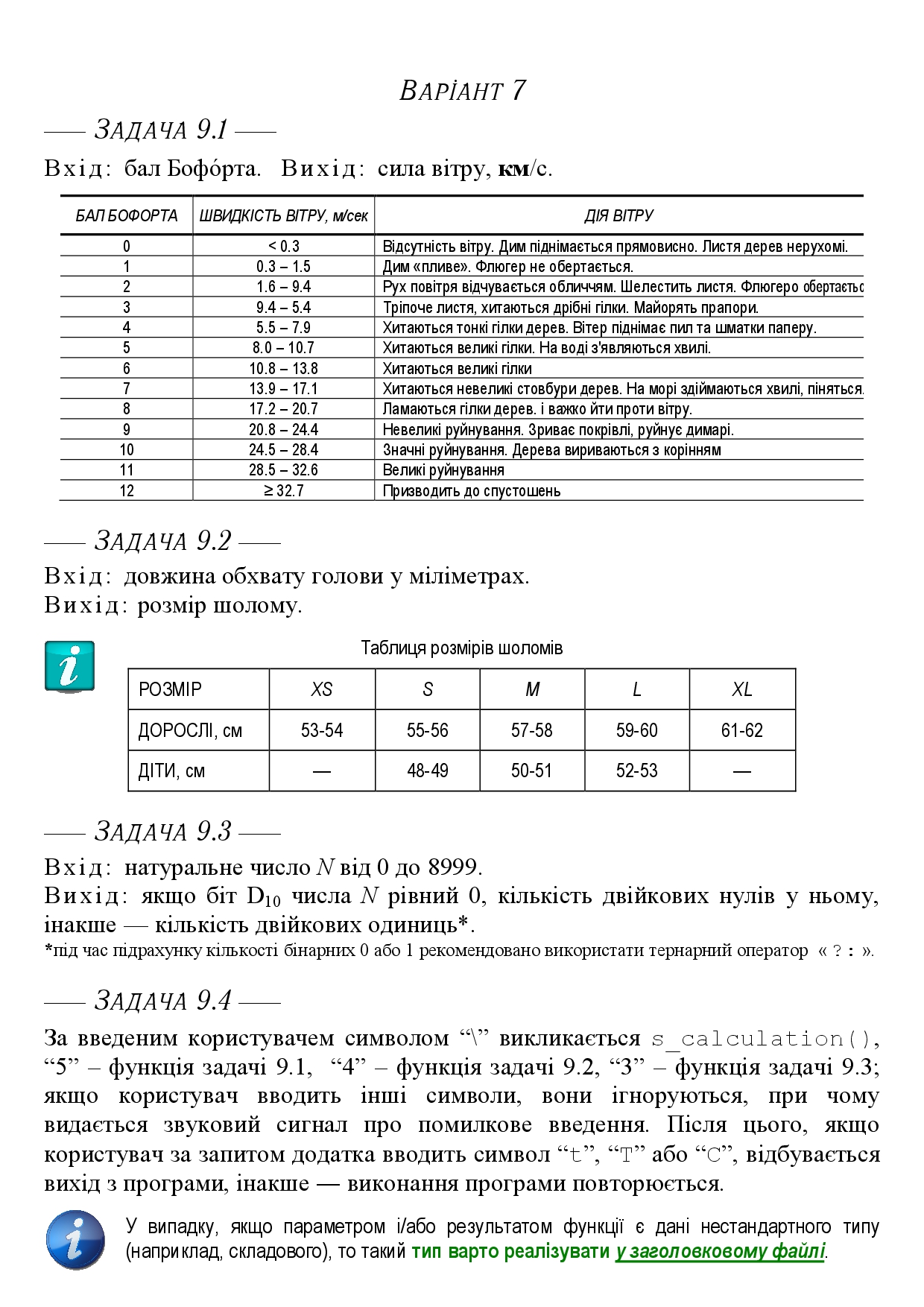
**Проектування архітектури програмного модуля**

Згідно з вимог проектування реалізуємо функцію, яка приймає на вхід ціле число. Повертає рядок із значенням сили вітру.

Якщо введене користувачем число не входить в таблицю Бофорта (невірно казаний бал), повертає рядок із повідомленням про невірно введене значення.

Якщо число входить в таблицю Бофорта (правильно вказано бал), повертає силу вітру із заздалегідь переведеними значеннями (у таблиці сила вітру вимірюється у м/с, було переведено у км/с).

Рисунок 1.2 – Умова задачі 9.2



**Строга постановка задачі**

Вхідні дані:

* Довжина обхвату голови у міліметрах

Вихідні дані:

* Розмір шолому

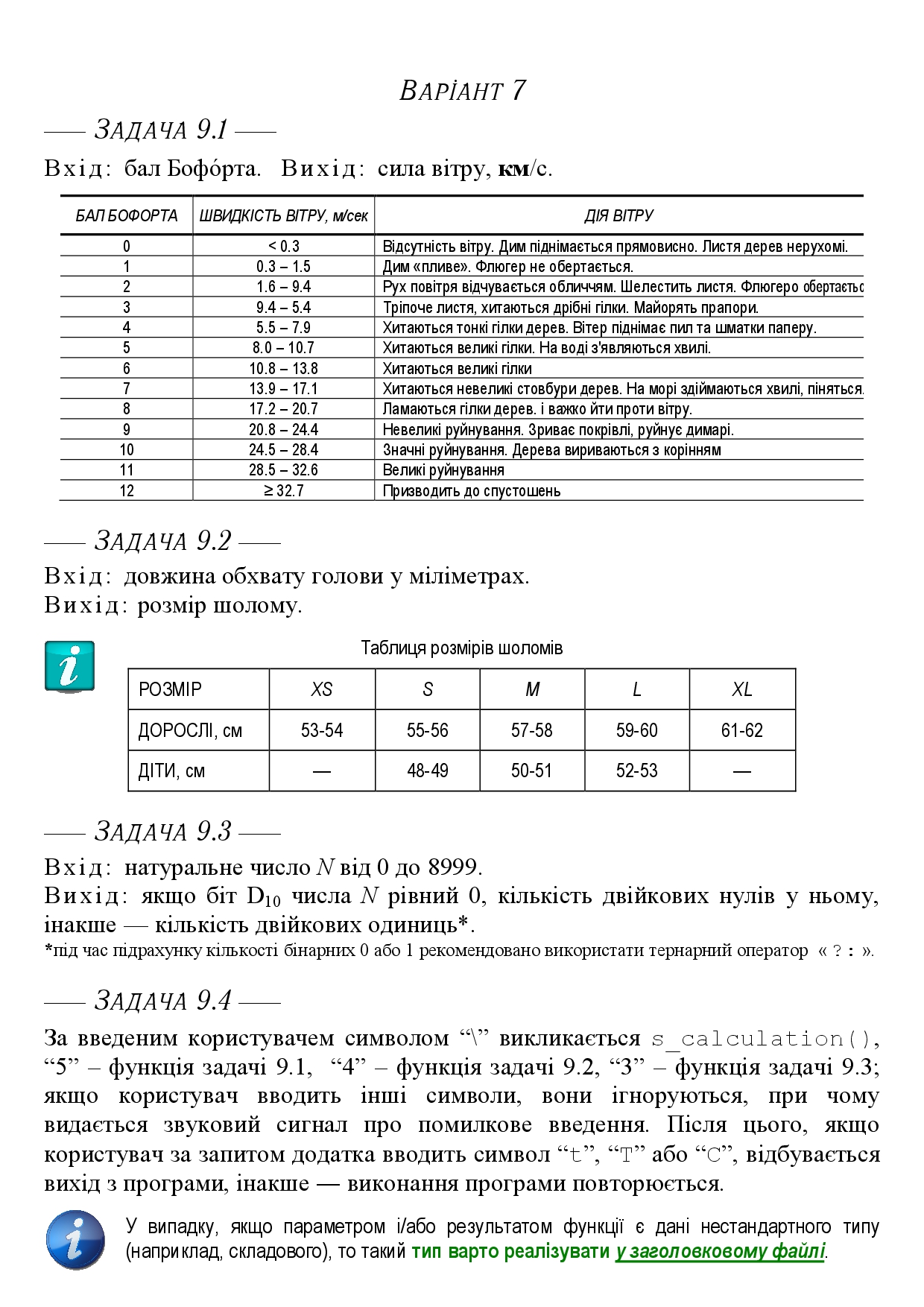
**Проектування архітектури програмного модуля**

Функція приймає на вхід ціле число – довжину обхвату голови у міліметрах і повертає рядок із повідомленням, чи це дитячий шолом чи дорослий і розмір шолому.

Якщо користувач вводить невірний розмір, повертає рядок із повідомленням про те, що в таблиці відсутній заданий розмір.

Якщо користувач вводить правильний розмір, введене число за допомогою функції заокруглюється до десятків (вхідні дані у мм, дані в таблиці у см).

Рисунок 1.3 – Умова задачі 9.3



**Строга постановка задачі**

Вхід:

* Натуральне число від 0 до 8999

Вихід:

* К-ть двійкових нулів якщо 10 біт = 0, інакше – к-ть двійкових одиниць

**Проектування архітектури програмного модуля**

Реалізована процедура приймає в якості аргументу число та повертає кількість двійкових нулів у випадку D10 = 0, інакше – к-ть двійкових одиниць.

Якщо користувач вводить значення, що не задовольняє проміжки, виводиться рядок із повідомленням про невірно ведене число, і саме число (щоб користувач мав можливість порівняти введене число із проміжком).

Далі, для подальшого опрацювання числа, воно ініціалізується оберненим. Створюємо додаткову змінну iter\_num для ітерацій циклу, яка приймає значення оберненого числа та змінну binZeros, яка відповідатиме за к-ть двійкових нулів. У циклі відбувається двійкове множення iter\_num і віднімання на 1 а також збільшення на один binZeros. За допомогою оператору множинного вибору, що опрацьовує значення 10 біта, у відповідності до цього ж значення і обирається виведення кількості необхідних бітів.

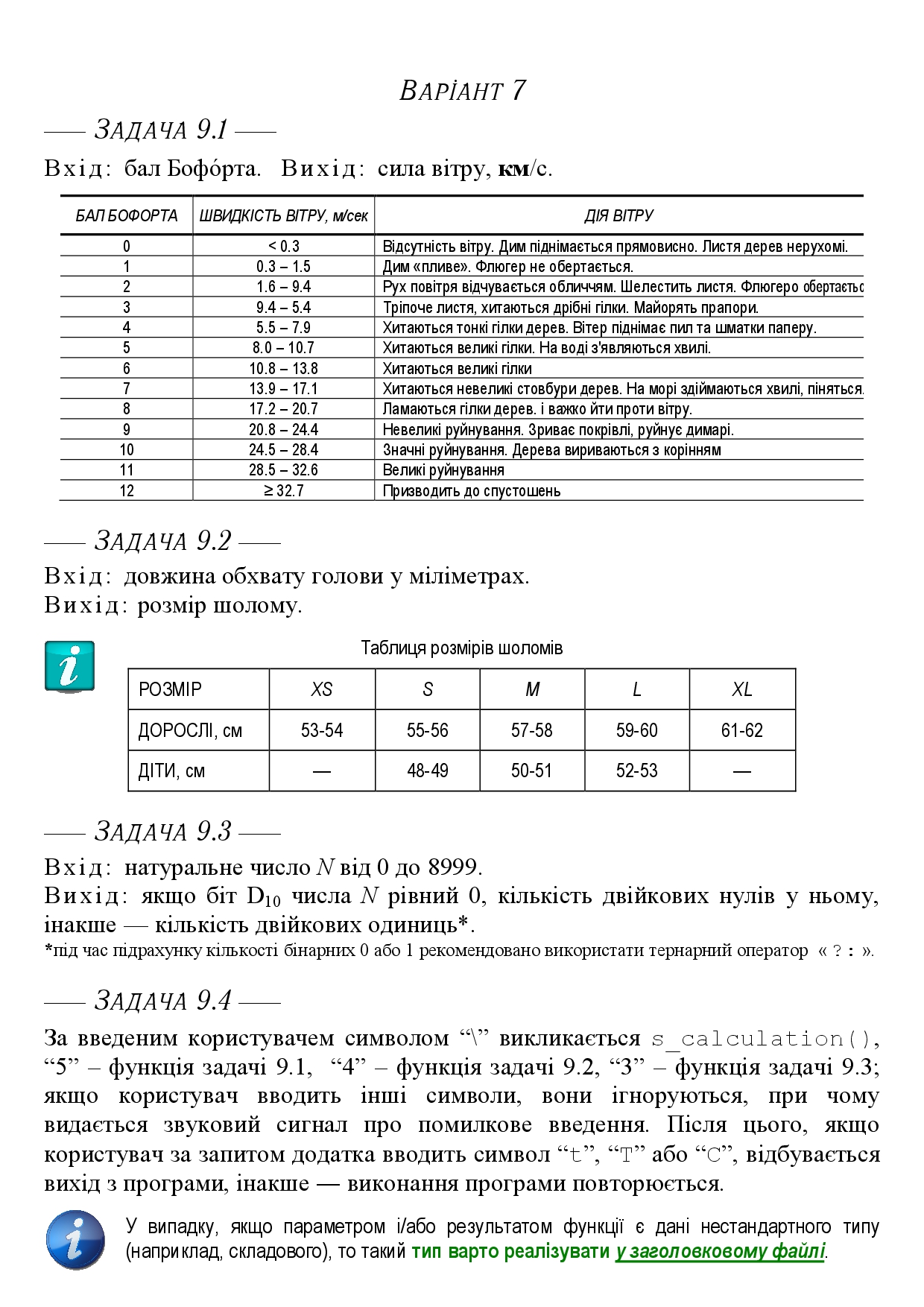


Рисунок 1.4 – Умова задачі 9.4

**Строга постанова задачі**

Вхідні дані:

* Символьний літерал, що вводить користувач

Вихідні дані:

* Результат виконання, відповідної до виклику літералом, функції

**Проектування архітектури програмного модуля**

На початку роботи програмного забезпечення виводимо анотацію, де вказано, який літерал за яку функцію відповідає. Використовуючи оператор множинного вибору, що обробляє значення, введене з клавіатури, реалізуємо введення користувачем даних та подальший виклик функцій, у які передаються введені дані. Шляхи виконання визначені в умові, як і літерал, введення якого обумовлює вихід з вічного циклу.

**Результат запуску застосунку TestDriver**

=======================================================================================

TEST SUITE 9.1

TEST CASE #1

Incoming data:

Beaufort Score = -1

Expected result: Invalid value

Actual result: Invalid value

Test case #1 PASSED

TEST CASE #2

Incoming data:

Beaufort Score = 0

Expected result: Wind speed less than 0.0003 km/s

Actual result: Wind speed less than 0.0003 km/s

Test case #2 PASSED

TEST CASE #3

Incoming data:

Beaufort Score = 7

Expected result: Wind speed varies from 0.0139 to 0,0171 km/s

Actual result: Wind speed varies from 0.0139 to 0,0171 km/s

Test case #3 PASSED

TEST CASE #4

Incoming data:

Beaufort Score = 12

Expected result: Wind speed more than 0.0327 km/s

Actual result: Wind speed more than 0.0327 km/s

Test case #4 PASSED

TEST CASE #5

Incoming data:

Beaufort Score = 13

Expected result: Invalid value

Actual result: Invalid value

Test case #5 PASSED

=======================================================================================

=======================================================================================

TEST SUITE 9.2

TEST CASE #1

Incoming data:

Head size = 474

Expected result: Your size is not in the table

Actual result: Your size is not in the table

Test case #1 PASSED

TEST CASE #2

Incoming data:

Head size = 475

Expected result: Children's helmet, size S

Actual result: Children's helmet, size S

Test case #2 PASSED

TEST CASE #3

Incoming data:

Head size = 543

Expected result: Adult's helmet, size XS

Actual result: Adult's helmet, size XS

Test case #3 PASSED

TEST CASE #4

Incoming data:

Head size = 624

Expected result: Adult's helmet, size XL

Actual result: Adult's helmet, size XL

Test case #4 PASSED

TEST CASE #5

Incoming data:

Head size = 625

Expected result: Your size is not in the table

Actual result: Your size is not in the table

Test case #5 PASSED

=======================================================================================

=======================================================================================

TEST SUITE 9.3

TEST CASE #1

Incoming data:

Beaufort Score = -1

Expected result: Wrong number! Your number: -1

Actual result: Wrong number! Your number: -1

Test case #1 PASSED

TEST CASE #2

Incoming data:

Beaufort Score = 511

Expected result: Number of binary 0 = 23

Actual result: Number of binary 0 = 23

Test case #2 PASSED

TEST CASE #3

Incoming data:

Beaufort Score = 1023

Expected result: Number of binary 1 = 10

Actual result: Number of binary 1 = 10

Test case #3 PASSED

TEST CASE #4

Incoming data:

Beaufort Score = 8999

Expected result: Number of binary 1 = 7

Actual result: Number of binary 1 = 7

Test case #4 PASSED

TEST CASE #5

Incoming data:

Beaufort Score = 9000

Expected result: Wrong number! Your number: 9000

Actual result: Wrong number! Your number: 9000

Test case #5 PASSED

=======================================================================================

**ВИСНОВКИ**

Під час виконання лабораторної роботи можна було помітити, до розв’язування були представлені задачі, взяті із попередніх лабораторних робіт, проте з використанням модульної парадигми програмування та більш широкого спектру взаємодії із середовищем розробки. Керуючись строгою постановкою було вирішено проблему некоректного вводу даних: у разі некоректного набору кожна функція повертає повідомлення про невірно введені дані (функція задачі 9.1 повертає повідомлення про некоректний ввід, функція задачі 9.2 повертає повідомлення про відсутність розміру в таблиці, функція задачі 9.3 повертає повідомлення про некоректний ввід).

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 задачі 9.1 було прийнято декілька проектних рішень. По-перше, для пришвидшення роботи програмного застосунку, було прийняти рішення запобігти обчислення швидкості вітру: швидкість була попередньо переведена з м/с у км/с і записана як частина рядкового літералу при повертанні значення в операторі return. Повертання значення реалізовано через оператор множинного вибору switch (13 кейсів повертають значення сили вітру, default повертає повідомлення про помилку).

Аналогічно до задачі 9.1, задача 9.2 повертає рядок із повідомленням, чи дорослий/дитячий це шолом і його розмір. У випадку, якщо отриманій розмір (в якості аргументу) не відноситься до жодного до зазначених розмірів у таблиці, функція повертає повідомлення про те, що в таблиці немає вказаного розміру. Проектне рішення, щодо оптимізації коду для даної задачі: за таблицею відповідників розмірів була визначена закономірність і з метою запобігання надлишкових рядків коду, розмір виводився із створеного рядкового масиву.

Передбачено отримання хибних вхідних даних і для функції задачі 9.3: функція повертає повідомлення про невірне значення числа. Надалі реалізовано фрагмент коду, що безпосередньо відповідає за побітові маніпуляції: щоб запобігти видозміні числа, було створену додаткову змінну iter\_num, яка прирівнялась до оберненого числа. Також був створений лічильник binZeros, який відповідає за підрахунок двійкових нулів. Далі всередині циклу, кількість ітерацій якого визначається кількістю бітів числа (тобто 32), відбувається двійкове множення числа саме на себе та віднімання від нього одиниці, також відбувається збільшення binZeros на 1. Виведення результату реалізовано за допомогою оператору множинного вибору switch, що перевіряє значення 10 біту. Задля оптимізації коду і використання меншої кількості пам’яті, лічильник, який відповідав би за підрахунок одиниць не було створено. Замість цього за потреби вивести кількість двійкових одиниць, від 32 віднімалася к-ть двійкових нулів, адже 32 = к-ть двійк. нулів + к-ть двійк. одиниць, а отже к-ть двійк. одиниць = 32 - к-ть двійк. нулів.

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 значення, що вводить користувач за запитом. 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 просто із 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 результату повернення використовують рядки.

**Додаток А**

(TestSuite до завдання 9.1)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_9\_1 |
| Date: 24/03/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.exe |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Барамба Андрій |
|  | Виконавець /  **Implementer** | Барамба Андрій |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Incoming data** | **Expected Result** | **Test Result** |
| TC-01 | -1 | Вікно застосунку:  "Invalid value" | Passed |
| TC-02 | 0 | Вікно застосунку:  "Wind speed less than 0.0003 km/s" | Passed |
| TC-03 | 7 | Вікно застосунку:  "Wind speed varies from 0.0139 to 0,0171 km/s" | Passed |
| TC-04 | 12 | Вікно застосунку:  "Wind speed more than 0.0327 km/s" | Passed |
| TC-05 | 13 | Вікно застосунку:  "Invalid value" | Passed |

**Додаток Б**

(TestSuite до завдання 9.2)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_9\_2 |
| Date: 24/03/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.exe |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Барамба Андрій |
|  | Виконавець /  **Implementer** | Барамба Андрій |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Incoming data** | **Expected Result** | **Test Result** |
| TC-01 | 474 | Вікно застосунку:  "Your size is not in the table" | Passed |
| TC-02 | 475 | Вікно застосунку:  "Children's helmet, size S" | Passed |
| TC-03 | 543 | Вікно застосунку:  "Adult's helmet, size XS" | Passed |
| TC-04 | 624 | Вікно застосунку:  "Adult's helmet, size XL" | Passed |
| TC-05 | 625 | Вікно застосунку:  "Your size is not in the table " | Passed |

**Додаток В**

(TestSuite до завдання 9.3)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_9\_3 |
| Date: 24/03/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.exe |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Барамба Андрій |
|  | Виконавець /  **Implementer** | Барамба Андрій |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Incoming data** | **Expected Result** | **Test Result** |
| TC-01 | -1 | Вікно застосунку:  Wrong number!  Your number: -1 | Passed |
| TC-02 | 511 | Вікно застосунку:  Number of binary 0 = 23 | Passed |
| TC-03 | 1024 | Вікно застосунку:  Number of binary 1 = 10 | Passed |
| TC-04 | 8999 | Вікно застосунку:  Number of binary 1 = 7 | Passed |
| TC-05 | 9000 | Вікно застосунку:  Wrong number! Your number: 9000 | Passed |

**Додаток Г**

(вихідний код ModulesBaramba)

**… / ModulesBaramba / main.cpp**

#include <iostream>

#include "cmath"

using namespace std;

double s\_calculation(float x, float z)

{

const double pi = 3.14159;

return (sqrt(1 + x) - 3 \* cos(x)) / (log(pow(x, 2)) + 3 \* sin(pi \* x)) + pow(sqrt(pow(z, 3)) + 2, 2);

}

string BeaufortScale(int BeaufortScore) {

switch (BeaufortScore){

case 0:

return "Wind speed less than 0.0003 km/s";

case 1:

return "Wind speed varies from 0.0003 to 0.0015 km/s";

case 2:

return "Wind speed varies from 0.0016 to 0,0094 km/s";

case 3:

return "Wind speed varies from 0.0094 to 0,0054 km/s";

case 4:

return "Wind speed varies from 0.0055 to 0,0079 km/s";

case 5:

return "Wind speed varies from 0.008 to 0,0107 km/s";

case 6:

return "Wind speed varies from 0.0108 to 0,0138 km/s";

case 7:

return "Wind speed varies from 0.0139 to 0,0171 km/s";

case 8:

return "Wind speed varies from 0.0172 to 0,0207 km/s";

case 9:

return "Wind speed varies from 0.0208 to 0,0224 km/s";

case 10:

return "Wind speed varies from 0.0245 to 0,0284 km/s";

case 11:

return "Wind speed varies from 0.0285 to 0,0326 km/s";

case 12:

return "Wind speed more than 0.0327 km/s";

default:

return "Invalid value";

}

}

string helmetSize(int headSize){

if(headSize < 475 || headSize > 624){

return "Your size is not in the table";

}

headSize = round(headSize / 10);

int minimalSize = 47;

char helmetSizes[8][255] = {"Children's helmet, size S",

"Children's helmet, size M",

"Children's helmet, size L",

"Adult's helmet, size XS",

"Adult's helmet, size S",

"Adult's helmet, size M",

"Adult's helmet, size L",

"Adult's helmet, size XL",

};

int i = 0;

for(; minimalSize < 63; minimalSize++){

if(minimalSize != headSize){

if(minimalSize % 2 == 0) i++;

continue;

} else {

return helmetSizes[i];

}

}

}

string binOnesZeros(int number){

int binZeros = 0;

int iter\_num = ~number;

while(iter\_num){

iter\_num &= iter\_num - 1;

binZeros ++;

}

if(number < 0 || number > 8999){

return "Wrong number! Your number: " + to\_string(number);

}

switch(number >> 9 & 1){

case 0:

return "Number of binary 0 = " + to\_string(binZeros);

case 1:

return "Number of binary 1 = " + to\_string(32 - binZeros);

}

}

**… / prj / ModulesBaramba.h**

#ifndef MODULESBARAMBA\_H\_INCLUDED

#define MODULESBARAMBA\_H\_INCLUDED

#include <iostream>

#include <cmath>

double s\_calculation(float x, float z);

std::string BeaufortScale(int BeaufortScore);

std::string helmetSize(int headSize);

std::string binOnesZeros(int number);

#endif // MODULESBARAMBA\_H\_INCLUDED

**Додаток Д**

(вихідний код TestDriver)

**… / TestDriver / main.cpp**

#include <iostream>

#include "ModulesBaramba.h"

using namespace std;

int main()

{

int BeaufortScoreValues[5] {-1, 0, 7 ,12, 13};

char BeaufortScaleResult[5][255] {"Invalid value",

"Wind speed less than 0.0003 km/s",

"Wind speed varies from 0.0139 to 0,0171 km/s",

"Wind speed more than 0.0327 km/s",

"Invalid value"};

int headSizeValues[5] {474, 475, 543, 624, 625};

char helmetSizeResult[5][255] {"Your size is not in the table",

"Children's helmet, size S",

"Adult's helmet, size XS",

"Adult's helmet, size XL",

"Your size is not in the table"};

int numberValues[5] {-1, 511, 1023, 8999, 9000};

char binOnesZerosResult[5][255] {"Wrong number! Your number: -1",

"Number of binary 0 = 23",

"Number of binary 1 = 10",

"Number of binary 1 = 7",

"Wrong number! Your number: 9000"};

cout << "=======================================================================================" << endl;

cout << "TEST SUITE 9.1" << endl;

for(int i = 0; i < 5; i++){

cout << "\tTEST CASE #"<< i + 1 << "\n"

"\t\t\tIncoming data: " << "\n"

"\t\t\tBeaufort Score = " << BeaufortScoreValues[i] << "\n";

cout << "\t\t\tExpected result: " << BeaufortScaleResult[i] << endl;

cout << "\t\t\tActual result: " << BeaufortScale(BeaufortScoreValues[i]) << endl;

if(BeaufortScale(BeaufortScoreValues[i]) == BeaufortScaleResult[i]){

cout << "\t\t\tTest case #" << i + 1 << " PASSED\n" << endl;

} else {

cout << "\t\t\tTest case #" << i + 1 << " FAILED\n" << endl;

}

}

cout << "=======================================================================================" << endl << endl;

cout << "=======================================================================================" << endl;

cout << "TEST SUITE 9.2" << endl;

for(int i = 0; i < 5; i++){

cout << "\tTEST CASE #"<< i + 1 << "\n"

"\t\t\tIncoming data: " << "\n"

"\t\t\tHead size = " << headSizeValues[i] << "\n";

cout << "\t\t\tExpected result: " << helmetSizeResult[i] << endl;

cout << "\t\t\tActual result: " << helmetSize(headSizeValues[i]) << endl;

if(helmetSize(headSizeValues[i]) == helmetSizeResult[i]){

cout << "\t\t\tTest case #" << i + 1 << " PASSED\n" << endl;

} else {

cout << "\t\t\tTest case #" << i + 1 << " FAILED\n" << endl;

}

}

cout << "=======================================================================================" << endl << endl;

cout << "=======================================================================================" << endl;

cout << "TEST SUITE 9.3" << endl;

for(int i = 0; i < 5; i++){

cout << "\tTEST CASE #"<< i + 1 << "\n"

"\t\t\tIncoming data: " << "\n"

"\t\t\tBeaufort Score = " << numberValues[i] << "\n";

cout << "\t\t\tExpected result: " << binOnesZerosResult[i] << endl;

cout << "\t\t\tActual result: " << binOnesZeros(numberValues[i]) << endl;

if(binOnesZeros(numberValues[i]) == binOnesZerosResult[i]){

cout << "\t\t\tTest case #" << i + 1 << " PASSED\n" << endl;

} else {

cout << "\t\t\tTest case #" << i + 1 << " FAILED\n" << endl;

}

}

cout << "=======================================================================================" << endl << endl;

}

**Додаток Е**

(TestSuite до завдання 9.4)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_9\_4 |
| Date: 25/03/2022 | Назва проекта/ПЗ /  **Name of project** | Baramba\_task.exe |
|  | Рівень тестування /  **Level of testing** | Системний |
|  | Автор тест-сьюта /  **Test Suite Author** | Барамба Андрій |
|  | Виконавець /  **Implementer** | Барамба Андрій |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Incoming data** | **Expected Result** | **Test Result** |
| TC-01 | 1. Запустити застосунок | У вікно виводиться анотація щодо значень які повинні водитися та за які функції/дії вони відповідають | Passed |
| TC-02 | 1. Запустити застосунок 2. Натиснути Enter | Вікно застосунку:  Enter symbol (\/5/4/3/t/T/C): | Passed |
| TC-03 | 1. Запустити застосунок 2. Натиснути Enter 3. Натиснути “\” 4. Ввести 5 5. Ввести 4 | Вікно застосунку:  Enter x value: 5  Enter z value: 4  S = 100.497  Enter symbol (\/5/4/3/t/T/C): | Passed |
| TC-04 | 1. Запустити застосунок 2. Натиснути Enter 3. Натиснути “5” 4. Ввести 7 | Вікно застосунку:  Enter Beaufort score: 7  Wind speed varies from 0.0139 to 0,0171 km/s  Enter symbol (\/5/4/3/t/T/C): | Passed |
| TC-05 | 1. Запустити застосунок 2. Натиснути Enter 3. Натиснути “4” 4. Ввести 567 | Вікно застосунку:  Enter head size (mm): 567  Adult's helmet, size S  Enter symbol (\/5/4/3/t/T/C): | Passed |
| TC-06 | 1. Запустити застосунок 2. Натиснути Enter 3. Натиснути “3” 4. Ввести 1023 | Вікно застосунку:  Enter number (0 - 8999): 1023  Number of binary 1 = 10  Enter symbol (\/5/4/3/t/T/C): | Passed |
| TC-07 | 1. Запустити застосунок 2. Натиснути Enter 3. Ввести “t” | Застосунок закривається | Passed |
| TC-08 | 1. Запустити застосунок 2. Натиснути Enter 3. Ввести “T” | Застосунок закривається | Passed |
| TC-09 | 1. Запустити застосунок 2. Натиснути Enter 3. Ввести “C” | Застосунок закривається | Passed |
| TC-10 | 1. Запустити застосунок 2. Натиснути Enter 3. Ввести “u” 4. Ввести “d” 5. Ввести “Z” 6. Ввести “Q” | Повторяється запит на введення, із відповідним звуковим сигналом | Passed |