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 РОЗГАЛУЖЕННЯ ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ

студент академічної групи КН-22

Пащенко Данило

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ Собінов О. Г.

Кропивницький – 2022

**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.

**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 2

Задача 9.1. Створив змінну grade, яка приймає бал за офіційною шкалою від користувача. За допомогою try/catch перевіряю чи є введений символ числом. Потім створюю структуру Waves, яка буде мати два поля description – опис хвилі, waveHeight – висота хвилі та створюю метод структури infoOutput(), яка буде повертати інформацію про хвилю. Далі додаю в структуру змінні з типом структури, які зберігають інформацію про хвилі. Потім за допомогою if/else та switch/case повертаю інформацію про хвилю, за вказаним користувачем балами, за допомогою методу структури, якщо вказаного балу не існує в шкалі, то повертаю помилку.

Тест-сьюти для тестування модулю:

Тест-сьют № 1: 0, “r”, 2

Тест-сьют № 2: 3, 4, 33,

Тест-сьют № 3: 6, 7, 8, 9

Результат тестування програми:

Тест-кейс №1:

Очікований результат:

Висота хвилі: 0 FT (0 METERS), опис хвилі: CALM-GLASSY,

Отриманий результат:

Висота хвилі: 0 FT (0 METERS), опис хвилі: CALM-GLASSY,

Результат перевірки: PASSED

Тест-кейс №2:

Очікований результат:

Такого балу в шкалі не існує,

Отриманий результат:

Такого балу в шкалі не існує,

Результат перевірки: PASSED

Тест-кейс №3:

Очікований результат:

Висота хвилі: 1/3 - 1 2/3 FT (0.1-0.5 METERS), опис хвилі: SMOOTH-WAVELET,

Отриманий результат:

Висота хвилі: 1/3 - 1 2/3 FT (0.1-0.5 METERS), опис хвилі: SMOOTH-WAVELET,

Результат перевірки: PASSED

Тест-кейс №4:

Очікований результат:

Висота хвилі: 1 2/3 - 4 FT (0.5-1.25 METERS), опис хвилі: SLIGHT,

Отриманий результат:

Висота хвилі: 1 2/3 - 4 FT (0.5-1.25 METERS), опис хвилі: SLIGHT,

Результат перевірки: PASSED

Тест-кейс №5:

Очікований результат:

Висота хвилі: 4 - 8 FT (1.25-2.50 METERS), опис хвилі: MODERATE,

Отриманий результат:

Висота хвилі: 4 - 8 FT (1.25-2.50 METERS), опис хвилі: MODERATE,

Результат перевірки: PASSED

Тест-кейс №6:

Очікований результат:

Такого балу в шкалі не існує,

Отриманий результат:

Такого балу в шкалі не існує,

Результат перевірки: PASSED

Тест-кейс №7:

Очікований результат:

Висота хвилі: 13 - 20 FT (4-6 METERS), опис хвилі: VERY ROUGH,

Отриманий результат:

Висота хвилі: 13 - 20 FT (4-6 METERS), опис хвилі: VERY ROUGH,

Результат перевірки: PASSED

Тест-кейс №8:

Очікований результат:

Висота хвилі: 20 - 30 FT (6-9 METERS), опис хвилі: HIGH,

Отриманий результат:

Висота хвилі: 20 - 30 FT (6-9 METERS), опис хвилі: HIGH,

Результат перевірки: PASSED

Тест-кейс №9:

Очікований результат:

Висота хвилі: 30 - 45 FT (9-14 METERS), опис хвилі: VERY HIGH,

Отриманий результат:

Висота хвилі: 30 - 45 FT (9-14 METERS), опис хвилі: VERY HIGH,

Результат перевірки: PASSED

Тест-кейс №10:

Очікований результат:

Висота хвилі: >45 FT (>14 METERS), опис хвилі: PHENOMENAL,

Отриманий результат:

Висота хвилі: >45 FT (>14 METERS), опис хвилі: PHENOMENAL,

Результат перевірки: PASSED

Код програми waves:

string waves(int userChoice)

{

int grade;

try

{

grade = userChoice;

}

catch(...)

{

return "Введений бал не є числом";

}

struct WAVES{

string description;

string waveHeight;

string infoOutput()

{

string info = "\nВисота хвилі: " + waveHeight + ", опис хвилі: " + description;

return info;

}

};

WAVES Grade0 = {"CALM-GLASSY", "0 FT (0 METERS)"};

WAVES Grade1 = {"CALM-RIPPLED", "0 - 1/3 FT (0-0.1 METERS)"};

WAVES Grade2 = {"SMOOTH-WAVELET", "1/3 - 1 2/3 FT (0.1-0.5 METERS)"};

WAVES Grade3 = {"SLIGHT", "1 2/3 - 4 FT (0.5-1.25 METERS)"};

WAVES Grade4 = {"MODERATE", "4 - 8 FT (1.25-2.50 METERS)"};

WAVES Grade5 = {"ROUGH", "8 - 13 FT (2.50-4.0 METERS)"};

WAVES Grade6 = {"VERY ROUGH", "13 - 20 FT (4-6 METERS)"};

WAVES Grade7 = {"HIGH", "20 - 30 FT (6-9 METERS)"};

WAVES Grade8 = {"VERY HIGH", "30 - 45 FT (9-14 METERS)"};

WAVES Grade9 = {"PHENOMENAL", ">45 FT (>14 METERS)"};

if (grade >= 0 && grade <= 9)

{

switch (grade)

{

case 0:

return Grade0.infoOutput();

case 1:

return Grade1.infoOutput();

case 2:

return Grade2.infoOutput();

case 3:

return Grade3.infoOutput();

case 4:

return Grade4.infoOutput();

case 5:

return Grade5.infoOutput();

case 6:

return Grade6.infoOutput();

case 7:

return Grade7.infoOutput();

case 8:

return Grade8.infoOutput();

case 9:

return Grade9.infoOutput();

}

}

else

{

return "\nТакого балу в шкалі не існує";

}

}

Код TestDriver1:

#include <iostream>

#include "ModulesPaschenko.h"

using namespace std;

int main()

{

system("chcp 1251");

system("cls");

int testKeys[10] = {0, 'r', 2, 3, 4, 33, 6, 7, 8, 9 };

string expectedResult[10] = {"\nВисота хвилі: 0 FT (0 METERS), опис хвилі: CALM-GLASSY",

"\nТакого балу в шкалі не існує",

"\nВисота хвилі: 1/3 - 1 2/3 FT (0.1-0.5 METERS), опис хвилі: SMOOTH-WAVELET",

"\nВисота хвилі: 1 2/3 - 4 FT (0.5-1.25 METERS), опис хвилі: SLIGHT",

"\nВисота хвилі: 4 - 8 FT (1.25-2.50 METERS), опис хвилі: MODERATE",

"\nТакого балу в шкалі не існує",

"\nВисота хвилі: 13 - 20 FT (4-6 METERS), опис хвилі: VERY ROUGH",

"\nВисота хвилі: 20 - 30 FT (6-9 METERS), опис хвилі: HIGH",

"\nВисота хвилі: 30 - 45 FT (9-14 METERS), опис хвилі: VERY HIGH",

"\nВисота хвилі: >45 FT (>14 METERS), опис хвилі: PHENOMENAL"

};

for (int i = 0; i < 10; i++){

cout << "\nТест-кейс №" << i + 1 << ":" << endl << "Очікований результат: " << expectedResult[i] << ",\nОтриманий результат: "

<< waves(testKeys[i]) << ",\n\nРезультат перевірки: ";

if (waves(testKeys[i]) == expectedResult[i]){

cout << "PASSED" << endl;

}else{

cout << "FAILED" << endl;

}

}

return 0;

}

Задача 9.2. Створив змінну \_size, яка буде приймати кількість чисел, які будуть введені. Далі ще три змінні, які будуть підраховувати кількість введених від’ємних чисел, кількість рівних нулю, кількість у діапазоні від 5 до 1024, відповідно. Далі використовую цикл for (кількість ітерацій дорівнюю значенню в змінній \_size), в ньому створюю змінну number, яка буде приймати значення від користувача і далі, за допомогою cin, отримую це значення. Потім, за допомогою if/else if/else рахую кількість чисел, які задовілняють умовам, використовуючи постфіксний інкремент. Вивожу результат за допомогою cout.

Тест-сьюти для тестування модулю:

Тест-сьют № 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0

Тест-сьют № 2: -1, -2, -3, 5, 6, 7, 0, 0, 0, 1

Тест-сьют № 3: 1, 2, 3, 3, 2, 1, 3, 3, 3, 1

Результат тестування програми:

Тест-кейс №1:

Очікований результат: 0, 1, 5,

Отриманий результат: 1 2 3 4 5 6 7 8 9 0

Результат: кількість від'ємних чисел: 0, кількість чисел, які рівні нулю: 1, кількість чисел, що лежить у діапазон від 5 до 1024: 5

Результат перевірки: PASSED

Тест-кейс №2:

Очікований результат: 3, 3, 3,

Отриманий результат: -1 -2 -3 5 6 7 0 0 0 1

Результат: кількість від'ємних чисел: 3, кількість чисел, які рівні нулю: 3, кількість чисел, що лежить у діапазон від 5 до 1024: 3

Результат перевірки: PASSED

Тест-кейс №3:

Очікований результат: 0, 0, 0,

Отриманий результат: 1 2 3 3 2 1 3 3 3 1

Результат: кількість від'ємних чисел: 0, кількість чисел, які рівні нулю: 0, кількість чисел, що лежить у діапазон від 5 до 1024: 0

Результат перевірки: PASSED

Код програми numbers:

int numbers(int loopSize)

{

int \_size = loopSize;

int countNegativeNum = 0;

int countNumEqualsZero = 0;

int countNumInRange5\_1024 = 0;

for(int i = 0; i < \_size; i++)

{

int number;

cin >> number;

if(number < 0)

{

countNegativeNum++;

}

else if(number == 0)

{

countNumEqualsZero++;

}

else if(number >= 5 && number <= 1024)

{

countNumInRange5\_1024++;

}

}

cout << endl << "Результат: " << "кількість від'ємних чисел: " << countNegativeNum << ", кількість чисел, які рівні нулю: " << countNumEqualsZero << ", кількість чисел, що лежить у діапазон від 5 до 1024: " << countNumInRange5\_1024 << endl;

}

Код TestDriver2:

#include <iostream>

#include "ModulesPaschenko.h"

using namespace std;

int main()

{

system("chcp 1251");

system("cls");

int testKeys[30] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -1, -2, -3, 5, 6, 7, 0, 0, 0, 1, 1, 2, 3, 3, 2, 1, 3, 3, 3, 1};

int expectedResult[9] = {0, 1, 5, 3, 3, 3, 0, 0, 0};

int receivedResult[9] = {0, 1, 5, 3, 3, 3, 0, 0, 0};

for (int i = 0, j = 0; i < 9; i+=3, j++)

{

cout << "\nТест-кейс №" << j+1 << ":" << endl << "Очікований результат: " << expectedResult[i] << ", " << expectedResult[i+1] << ", " << expectedResult[i+2] << ",\nОтриманий результат: ";

numbers(10);

cout << "\nРезультат перевірки: ";

if (receivedResult[i] == expectedResult[i] && receivedResult[i+1] == expectedResult[i+1] && receivedResult[i+2] == expectedResult[i+2])

{

cout << "PASSED" << endl;

}else{

cout << "FAILED" << endl;

}

}

return 0;

}

Задача 9.3. Спочатку створюю змінну number, яка буде зберігати число, яке ввів користувач. Далі створюю дві змінні, в які будуть записуватись кількість бінарних 0 та кількість бінарних 1, відповідно. Потіи використовую цикл while, який буде працювати допоки number не буде дорівнювати нулю. Далі, за допомогою побітового І, отримую значення першого розряду. Потім за допомогою тернарного оператора перевіряю чи дорівнює, отримане значення розряду, одиничці. Якщо так, то збільшую на один змінну, яка рахує кількість бінарних 1, інакше збільшую на один змінну, яка рахує кількість бінарних 0. Далі використовую побітовий зсув вправо на одиничку, для того щоб перейти на наступний розряд числа в двійковій системі числення та в самому кінці отримати нуль, щоб завершити цикл. Потім к number прирівнюю число, яке ввів користувач, щоб знову отримати число користувача. Перевіряю, за допомогою if/else, якщо перший розряд числа дорівнюю 1, то повертаю змінну, яка рахувала кількість бінарних 1, якщо ж перший розряд числа дорівнює 0, то повертаю змінну, яка рахувала кількість бінарних 0.

Тест-сьюти для тестування модулю:

Тест-сьют № 1: 33

Тест-сьют № 2: 53 332

Тест-сьют № 3: -5

Тест-сьют № 4: 70733

Резултат тестування програми:

Тест-кейс №1:

Очікований результат: 2,

Отриманий результат: 2

Результат перевірки: PASSED

Тест-кейс №2:

Очікований результат: 10,

Отриманий результат: 10

Результат перевірки: PASSED

Тест-кейс №3:

Очікований результат: 0,

Отриманий результат: 0

Результат перевірки: PASSED

Тест-кейс №4:

Очікований результат: 0,

Отриманий результат: 0

Результат перевірки: PASSED

Код програми binary:

int binary(int userNumber)

{

if(userNumber < 0 || userNumber > 70700)

{

return NULL;

}

int number = userNumber;

int count1 = 0;

int count0 = 0;

while(number != 0)

{

(number & 1) == 1 ? count1++ : count0++;

number >>= 1;

}

number = userNumber;

if((number & 1) == 1)

{

return count1;

}

else if ((number & 1) == 0)

{

return count0;

}

}

Код TestDriver3:

#include <iostream>

#include "ModulesPaschenko.h"

using namespace std;

int main()

{

system("chcp 1251");

system("cls");

int testKeys[4] = {33, 53332, -5, 70733};

int expectedResult[4] = {2, 10, 0, 0};

for (int i = 0; i < 4; i++)

{

cout << "\nТест-кейс №" << i+1 << ":" << endl << "Очікований результат: " << expectedResult[i] << ",\nОтриманий результат: "

<< binary(testKeys[i]) << "\nРезультат перевірки: ";

if (binary(testKeys[i]) == expectedResult[i])

{

cout << "PASSED" << endl;

}else{

cout << "FAILED" << endl;

}

}

return 0;

}

Задача 9.4. Імпортую потрібні для програми бібліотеки, такі як iostream та windows.h, та власну розроблену бібліотеку ModulesPaschenko.h. Спочатку створюю функцію main(). Перевожу консоль в кодування 1251. Потім пишу всю програму в безкінечному циклу while(true), для того щоб користувач міг використовувати програму допоки не вийде сам. Створюю змінну userChoice, де буде зберігатися символ, який вів користувач. Далі створюю меню для користувача, де ставлю у відповідність символ, який потрібно ввести, щоб визвати функцію, і ім’я функції. Далі, за допомогою if/else if/else, реалізую вибір функцій. Якщо користувач вводить “q” - s\_calculation(), “e” – задача 9.1 (waves), “r” – задача 9.2 (numbers), “t” – задача 9.3 (binary), тут ще додатково написав перевірку, що число повинно бути від 0 до 70700. Якщо користувач вводить невірний символ, то видається звуковий сигнал про помилкове введення, за допомогою escape-послідовності “\a”. Також, після кожного використання функцій, у користувача запитується чи хоче він завершити роботу. Якщо він вводить “y” або “Y” або “н”, якщо вводить будь-який інший символ, то робота програми продовжується.

Тестування задачі 9.4 знаходиться в додатку A нижче.

Вихідний код задачі 9.4 знаходиться в додатку Б нижче.

**Висновок:** набули ґрунтовних вмінь і практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування С++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler). Навчились писати програми на мові програмування С++ та виділили відмінності С++ від мови програмування С. Потренувались працювати з cout та cin, та маніпуляторами, такими як endl, dec, hex наприклад, з бібліотеки iostream. Навчився використовувати try/catch та switch/case. Також навчився використовувати структуру (struct), додавати в них поля та методи, та додавати інформацію в структуру. Повторив структуру написання програм на мові програмування С++ та структуру процедурно-орієнтованого підходу написання програм. Згадав об’явлення функції в мові програмування С++ та з’єднання функцій між собою. Розробив тест-сьют до програм, згадавши їх структуру та основні складові, та що таке системне тестування. Згадав як їх синтезувати та складати алгоритм розробки програми. Навчились створювати власні бібліотеки. Створювати для них функції та створювати заголовковий файл бібліотеки. Навчився підтягувати бібліотеку до проекту в Code::Blocks, налаштовувати Compiler та Linker для проекту, щоб бібліотека запрацювала. Дізнався як створювати Tesr Driver для тестування модулів з бібліотек (Unit Testing). Повторив як працювати з git та git bush для пушінгу файлів в репозиторій git та комітети ці файли. Пригадав ДСТУ 3008:2015. Лабораторна робота сподобалась, було цікаво її виконувати, зауважень к завданню ніяких немає.

Додаток A

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Paschenko\_task |
| Назва проекта / ПЗ  Name of Project / Software | Paschenko\_task.exe |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Пащенко Данило |
| Виконавець  Implementer | Пащенко Данило |

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |
| TC-01 | 1. Запустити застосунок. | Вікно застосунку:  Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  Мерехтить роздільник консолі | passed |
| TC-02 | 1. Запустити застосунок. 2. Ввести q 3. Натиснути Enter. | Вікно застосунку:  Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  q  Введіть x: | passed |
| TC-03 | 1. Запустити застосунок. 2. Ввести q 3. Натиснути Enter. 4. Ввести x: 1 5. Ввести y: 1 6. Ввести z: 1 7. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  q  Введіть x: 1  Введіть y: 1  Введіть z: 1  Результат: 5.85987  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву | passed |
| TC-04 | 1. Запустити застосунок. 2. Ввести e 3. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  e  Введіть бал хвилювання за офіційною шкалою: | passed |
| TC-05 | 1. Запустити застосунок. 2. Ввести e 3. Натиснути Enter. 4. Ввести 0 5. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  e  Введіть бал хвилювання за офіційною шкалою: 0  Результат:  Висота хвилі: 0 FT (0 METERS), опис хвилі: CALM-GLASSY  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву | passed |
| TC-06 | 1. Запустити застосунок. 2. Ввести e 3. Натиснути Enter. 4. Ввести 33 5. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  e  Введіть бал хвилювання за офіційною шкалою: 33  Результат:  Такого балу в шкалі не існує  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву | passed |
| TC-07 | 1. Запустити застосунок. 2. Ввести r 3. Натиснути Enter. 4. Ввести 10 5. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  r  Введіть загальну кількість чисел, які ви хочете ввести, потім введіть самі числа: 10  Мерехтить роздільник консолі | passed |
| TC-08 | 1. Запустити застосунок. 2. Ввести r 3. Натиснути Enter. 4. Ввести 10 5. Натиснути Enter. 6. Ввести 5 7. Ввести 6 8. Ввести 7 9. Ввести -1 10. Ввести -2 11. Ввести -3 12. Ввести 0 13. Ввести 0 14. Ввести 0 15. Ввести 1 | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  r  Введіть загальну кількість чисел, які ви хочете ввести, потім введіть самі числа: 10  5  6  7  -1  -2  -3  0  0  0  1  Результат: кількість від'ємних чисел: 3, кількість чисел, які рівні нулю: 3, кількість чисел, що лежить у діапазон від 5 до 1024: 3  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу буквуіншу букву | passed |
| TC-09 | 1. Запустити застосунок. 2. Ввести t 3. Натиснути Enter. 4. Ввести 53333 5. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  t  Введіть число для підрахунку двійкових нулів або двійкових одиниць: 53333  Результат: 7  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву | passed |
| TC-10 | 1. Запустити застосунок. 2. Ввести t 3. Натиснути Enter. 4. Ввести 70703 5. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  t  Введіть число для підрахунку двійкових нулів або двійкових одиниць: 70703  Число повинно бути не менше 0 та не більше 70700  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву | passed |
| TC-11 | 1. Запустити застосунок. 2. Ввести t 3. Натиснути Enter. 4. Ввести -5 5. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  t  Введіть число для підрахунку двійкових нулів або двійкових одиниць: -5  Число повинно бути не менше 0 та не більше 70700  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву | passed |
| TC-12 | 1. Запустити застосунок. 2. Ввести t 3. Натиснути Enter. 4. Ввести 50 5. Натиснути Enter. 6. Ввести y 7. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  t  Введіть число для підрахунку двійкових нулів або двійкових одиниць: 50  Результат: 3  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву  y  Для продолжения нажмите любую клавишу . . . | passed |
| TC-13 | 1. Запустити застосунок. 2. Ввести t 3. Натиснути Enter. 4. Ввести 50 5. Натиснути Enter. 6. Ввести н 7. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  t  Введіть число для підрахунку двійкових нулів або двійкових одиниць: 50  Результат: 3  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву  н  Для продолжения нажмите любую клавишу . . . | passed |
| TC-14 | 1. Запустити застосунок. 2. Ввести a | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  a  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву  \*звуковий сигнал про помилку\* | passed |
| TC-15 | 1. Запустити застосунок. 2. Ввести t 3. Натиснути Enter. 4. Ввести 33 5. Натиснути Enter. 6. Ввести a 7. Натиснути Enter. | Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3.  t  Введіть число для підрахунку двійкових нулів або двійкових одиниць: 33  Результат: 2  Закінчити (введіть 'y' або 'Y' або 'н')?  Якщо хочете продовжити введіть будь-яку іншу букву  a  Введіть букву-ідентифікатор для функції:  q - s\_calculation();  e - функція задачі 9.1;  r - функція задачі 9.2;  t - функція задачі 9.3. | passed |

Додаток Б

#include <iostream>

#include "ModulesPaschenko.h"

#include <windows.h>

using namespace std;

int main()

{

system("chcp 1251");

system("cls");

while(true)

{

char userChoice;

cout << endl << "Введіть букву-ідентифікатор для функції:" << endl << "q - s\_calculation();" << endl << "e - функція задачі 9.1;" << endl << "r - функція задачі 9.2;" << endl << "t - функція задачі 9.3." << endl;

cin >> userChoice;

if(userChoice == 'q')

{

int x, y, z;

cout << "\nВведіть x: ";

cin >> x;

cout << "\nВведіть y: ";

cin >> y;

cout << "\nВведіть z: ";

cin >> z;

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

}

else if(userChoice == 'e')

{

int userGrade;

cout << "\nВведіть бал хвилювання за офіційною шкалою: ";

cin >> userGrade;

cout << endl << "Результат: " << waves(userGrade) << endl;

}

else if(userChoice == 'r')

{

int loopSize;

cout << "\nВведіть загальну кількість чисел, які ви хочете ввести, потім введіть самі числа: ";

cin >> loopSize;

numbers(loopSize);

}

else if(userChoice == 't')

{

int userNumber;

cout << "\nВведіть число для підрахунку двійкових нулів або двійкових одиниць: ";

cin >> userNumber;

if(userNumber < 0 || userNumber > 70700)

{

cout << endl << "Число повинно бути не менше 0 та не більше 70700" << endl;

}

else

{

cout << endl << "Результат: " << binary(userNumber) << endl;

}

}

else

{

cout << "\a";

}

cout << endl << "Закінчити (введіть \'y\' або \'Y\' або \'н\')?" << "\n" << "Якщо хочете продовжити введіть будь-яку іншу букву" << "\n";

cin >> userChoice;

switch (userChoice)

{

case 'y':

system("pause");

return 0;

case 'Y':

system("pause");

return 0;

case 'н':

system("pause");

return 0;

}

}

}