Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

# 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

КІ-21-1 Василенко Д.М.

Перевірив асистент

Усік П.С \_\_\_\_\_\_\_\_\_\_\_

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.

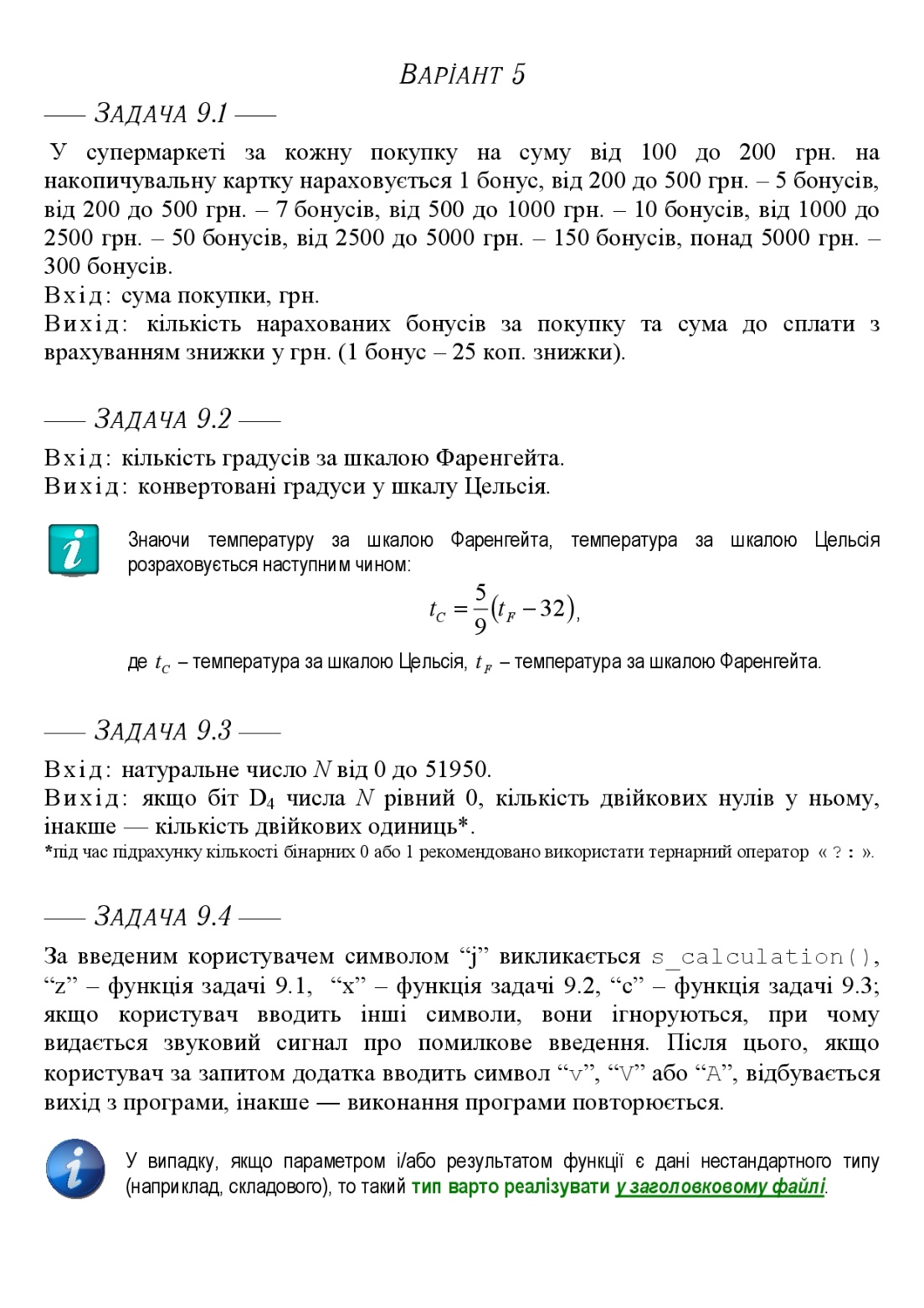


Рисунок 1.1 – Задачі варіанту № 5

**Формалізація задачі 9.1**

**Вхідні дані:**

*price* – сума покупки у гривнях, price ∈ [0; 9223372036854775808.0]

**Вихідні дані:**

*bonus* – кількість нарахованих бонусів за покупку, bonus = 0, , bonus = 1, bonus = 5, , bonus = 10, , bonus = 50, , bonus = 150, , bonus = 300;

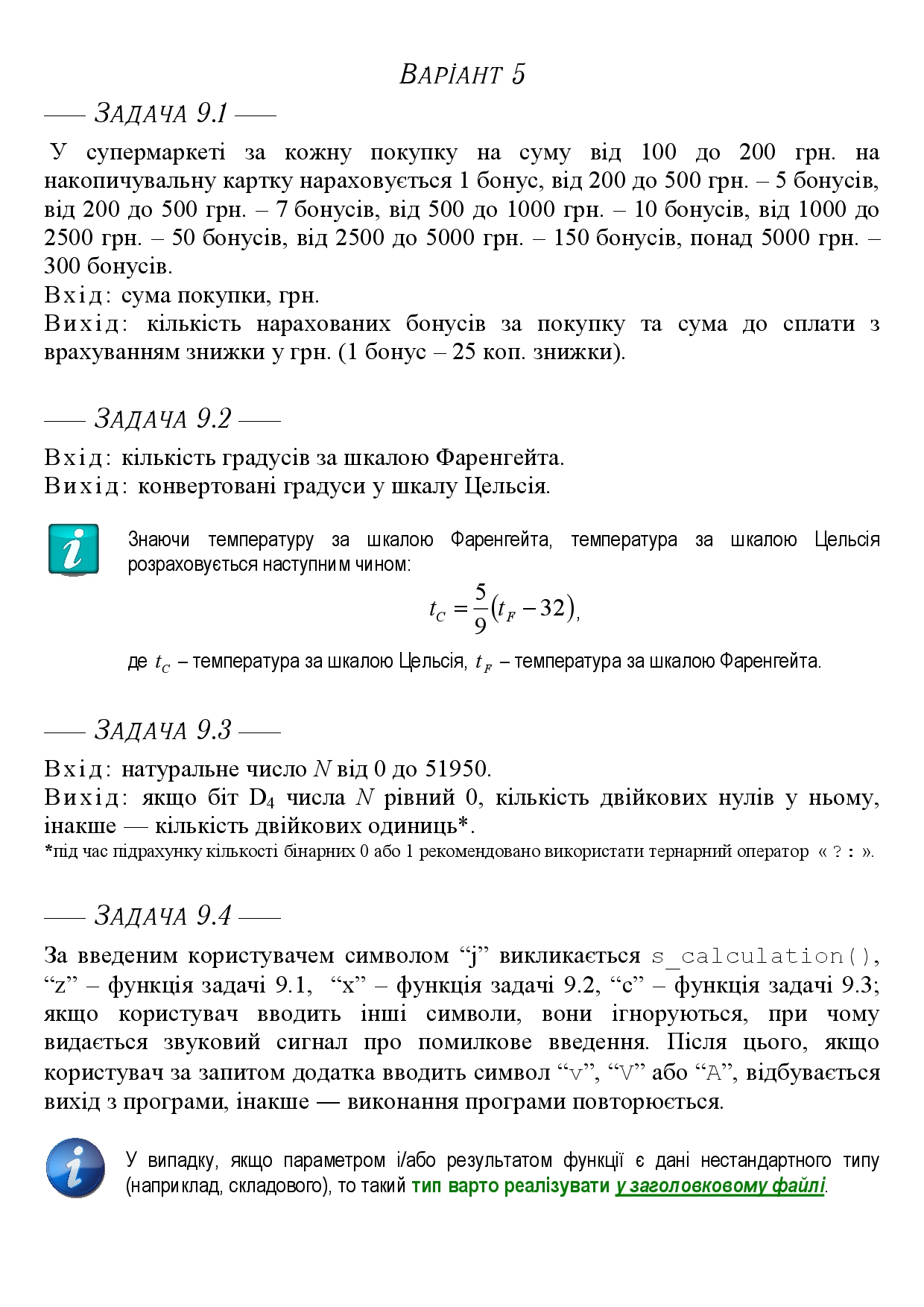
*pay* – сума до сплати з врахуванням знижки,

**Формалізація задачі 9.2**

**Вхідні дані:**

*t* – кількість градусів у шкалі Фаренгейта,

Математична модель:



**Вихідні дані:**

* Конвертовані градуси у шкалу Цельсія.

**Формалізація задачі 9.3**

**Вхідні дані:**

*N* – натуральне ціле число, що вводить користувач

**Вихідні дані:**

*result* – кількість двійкових одиниць або нулів у числі, що вводить користувач

Результат тестування:

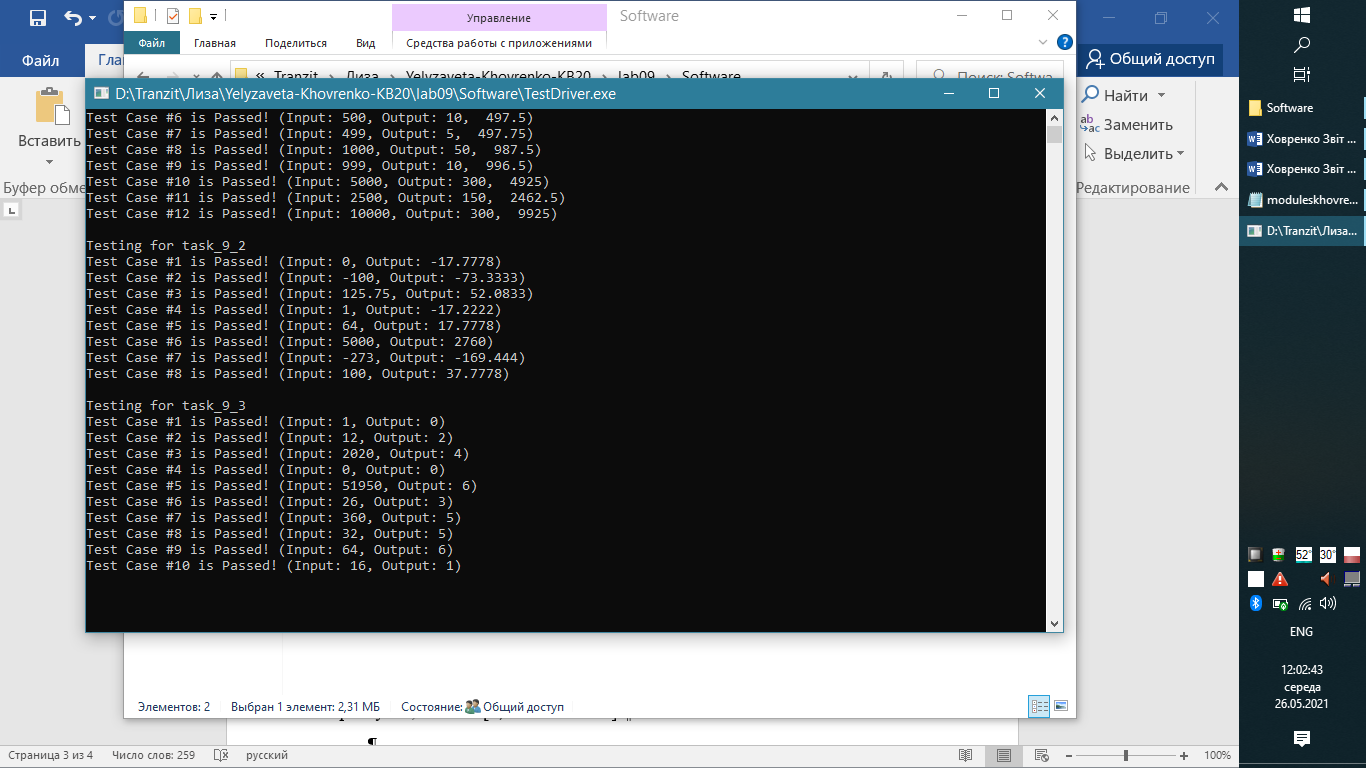


Рис. 2 - Результат тестування функцій зі статичної бібліотеки

**Формалізація задачі 9.4**

**Вхідні дані:**

*command* – літера, що вводить користувач щоб обрати функцію, яка буде виконуватись;

*symbol* – літера, що вводить користувач для завершення або повтору виконання програми;

*х, y*, *z* – аргументи для функції s\_calculation;

*price* – аргумент для функції задачі 9.1;

*t* – аргумент для функції задачі 9.2;

*n* - аргумент для функції задачі 9.3;

**Вихідні дані:**

1. Результат виконання функції s\_calculation;
2. Результат виконання функції задачі 9.1;
3. Результат виконання функції задачі 9.2;
4. Результат виконання функції задачі 9.3;

**Висновок:**

У ході виконання даної лабораторної роботи були реалізовані функції розв’язування задач 9.1 – 9.3 як складових статичної бібліотеки libModulesVasylenko.a (проект ModulesVasylenko, створений під час виконання лабораторної роботи №8) та було реалізовано програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesVasylenko.a.

По-перше, було виконано аналіз вимог, постановка задач, проектування архітектури та детальне проектування програмних модулів розв’язування задач 9.1 – 9.3. Після цього розроблено три набори контрольних прикладів до задач 9.1 – 9.3 для виконання модульного тестування одержаних функцій (див. Додатки А, Б, В).

Потім була виконана реалізація функцій мовою програмування С++ за отриманими під час проектування програмних модулів артефактами. Проект статичної бібліотеки ModulesVasylenko, який вже містить реалізовану функцію s\_calculation, доповнено функціями, які за наданим інтерфейсом реалізують розв’язування задач 9.1, 9.2 та 9.3 (див. Додаток Г). Проект заголовкового файлу ModulesVasylenko доповнено прототипами (заголовками) функцій розв’язування задач 9.1 -9.3.

Реалізовано тестовий драйвер для виконання модульного тестування функцій розв’язування задач 9.1, 9.2 та 9.3 за допомогою розроблених наборів контрольних прикладів. Реалізовано протоколювання процесу тестування тестовим драйвером: виводяться вхідні дані , отриманий від функції результат та статус кожного тест-кейса.

За допомогою розробленого тестового драйвера виконано модульне тестування функцій розв’язування задач 9.1 – 9.3 з бібліотеки libModulesVasylenko.a. (див. Додаток Д) Результат виконання всіх тест-кейсів – passed, що означає, що модуль працює згідно поставленим умовам.

Здійснено аналіз вимог до програмного забезпечення , постановка задачі, проектування архітектури та детальне проектування програмного забезпечення розв’язування задачі 9.4. Розроблено тест-сьют, що складається з 8 тест-кейсів, для виконання системного тестування програмного забезпечення для розв’язування задачі 9.4 (див. Додаток Е).

Створено проект консольного застосунку , у якому мовою програмування С++ реалізовані артефакти проектування програмного забезпечення розв’язування задачі 9.4 з використання функцій статичної бібліотеки libModulesVasylenko.a (див. Додаток Є)

Виконано системне тестування створеного додатку за допомогою розробленого тестового набору. Результат виконання всіх тест-кейсів є позитивним, що означає, що додаток коректо працює згідно поставлених завдань.

В результаті виконання лабораторної роботи мною були набутті ґрунтовні вміння і практичні навички 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 (GNU GCC Compiler).

**Додаток А**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Test-Suite9.1 |
| Назва проекта / ПЗ  Name of Project / Software | Vasylenko-task.exe |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Vasylenko Dmytro |
| Виконавець  Implementer | Vasylenko Dmytro |

|  |  |  |
| --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Вхід /  Input | Вихід /  Output |
| TC-1 | 0 | 0  0 |
| TC-2 | 100 | 1  99,75 |
| TC-3 | 200 | 5  198,75 |
| TC-4 | 199 | 1  198,75 |
| TC-5 | 99,99 | 0  99,99 |
| TC-6 | 500 | 10  497,50 |
| TC-7 | 499 | 5  497,75 |
| TC-8 | 1000 | 50  987,50 |
| TC-9 | 999 | 10  996,50 |
| TC-10 | 5000 | 300  4925,00 |
| TC-11 | 2500 | 150  2462,50 |
| TC-12 | 10000 | 300  9925,00 |

**Додаток Б**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Test-Suite9.2 |
| Назва проекта / ПЗ  Name of Project / Software | Vasylenko-task.exe |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Vasylenko Dmytro |
| Виконавець  Implementer | Vasylenko Dmytro |

|  |  |  |
| --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Вхід /  Input | Вихід /  Output |
| TC-1 | 0 | -17.7778 |
| TC-2 | -100 | -73,3333 |
| TC-3 | 125,75 | 52,0833 |
| TC-4 | 1 | -17,2222 |
| TC-5 | 64 | 17,7778 |
| TC-6 | 5000 | 2760 |
| TC-7 | -273 | -169,444 |
| TC-8 | 100 | 37,7778 |

**Додаток В**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Test-Suite9.3 |
| Назва проекта / ПЗ  Name of Project / Software | Vasylenko-task.exe |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Vasylenko Dmytro |
| Виконавець  Implementer | Vasylenko Dmytro |

|  |  |  |
| --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Вхід /  Input | Вихід /  Output |
| TC-1 | 1 | 0 |
| TC-2 | 12 | 2 |
| TC-3 | 2020 | 4 |
| TC-4 | 0 | 0 |
| TC-5 | 51950 | 6 |
| TC-6 | 26 | 3 |
| TC-7 | 360 | 5 |
| TC-8 | 32 | 5 |
| TC-9 | 64 | 6 |
| TC-10 | 16 | 1 |

**Додаток Г**

#include "ModulesVasylenko.h"

#include <cmath>

using namespace std;

double s\_calculation(double x, double y, double z)

{

double S;

S = z + M\_PI\*((pow((2\*z + 1), 2) - sqrt(abs(y - 0.5\*z))/sqrt(cos(z + y\*z)+ pow(x, 2))));

return S;

}

task\_9\_1 Bonus9\_1(double price)

{

task\_9\_1 task;

if (price < 100)

task.bonus = 0;

else

if (price>=100 && price<200)

task.bonus = 1;

else

if (price>=200 && price<500)

task.bonus = 5;

else

if (price>=500 && price<1000)

task.bonus = 10;

else

if (price>=1000 && price<2500)

task.bonus = 50;

else

if (price>=2500 && price<5000)

task.bonus = 150;

else

task.bonus = 300;

task.pay = price - task.bonus\*0.25;

return task;

}

float Temperature9\_2(float t)

{

return (5\*(t-32))/9;

}

int Binary9\_3(int N)

{

int result = 0;

bool flag = false;

for (int n=sizeof(int)\*8-1; n>-1; n--)

{

if((N>>n)&1)

if(!flag)

flag = true;

if (flag)

{

if((N>>4)&1)

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

else

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

}

}

return result;

}

**Додаток Д**

#include <iostream>

#include <ModulesVasylenko.h>

#include <cmath>

using namespace std;

int main()

{

cout << "Testing for task\_9\_1" << endl;

double input\_9\_1[12] = {0, 100, 200, 199, 99.99, 500, 499, 1000, 999, 5000, 2500, 10000};

short output\_9\_1\_bonus[12] = {0, 1, 5, 1, 0, 10, 5, 50, 10, 300, 150, 300};

double output\_9\_1\_pay[12] = {0, 99.75, 198.75, 198.75, 99.99, 497.50, 497.75, 987.50, 996.50, 4925.00, 2462.50, 9925.00};

for (int i = 0; i<12; i++)

{

if (Bonus9\_1(input\_9\_1[i]).bonus == output\_9\_1\_bonus[i] && Bonus9\_1(input\_9\_1[i]).pay == output\_9\_1\_pay[i])

cout << "Test Case #" << i+1 << " is Passed! (Input: " << input\_9\_1[i] << ", Output: " << Bonus9\_1(input\_9\_1[i]).bonus << ", " << Bonus9\_1(input\_9\_1[i]).pay << ")" << endl;

else

cout << "Test Case #" << i+1 << " is Failed... (Input: " << input\_9\_1[i] << ", Output: " << Bonus9\_1(input\_9\_1[i]).bonus << ", " << Bonus9\_1(input\_9\_1[i]).pay << ")" << endl;

}

cout << "\nTesting for task\_9\_2" << endl;

float input\_9\_2[8] = {0, -100, 125.75, 1, 64, 5000, -273, 100};

float output\_9\_2[8] = {-17.7778, -73.3333, 52.0833, -17.2222, 17.7778, 2760, -169.444, 37.7778};

for (int i = 0; i<8; i++)

{

if (fabs(Temperature9\_2(input\_9\_2[i]) - output\_9\_2[i]) < 0.001)

//if (Temperature9\_2(input\_9\_2[i]) == output\_9\_2[i])

cout << "Test Case #" << i+1 << " is Passed! (Input: " << input\_9\_2[i] << ", Output: " << output\_9\_2[i] << ")" << endl;

else

cout << "Test Case #" << i+1 << " is Failed... (Input: " << input\_9\_2[i] << ", Output: " << output\_9\_2[i] << ")" << endl;

}

cout << "\nTesting for task\_9\_3" << endl;

int input\_9\_3[10] = {1, 12, 2020, 0, 51950, 26, 360, 32, 64, 16};

int output\_9\_3[10] = {0, 2, 4, 0, 6, 3, 5, 5, 6, 1};

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

{

if (Binary9\_3(input\_9\_3[i]) == output\_9\_3[i])

cout << "Test Case #" << i+1 << " is Passed! (Input: " << input\_9\_3[i] << ", Output: " << output\_9\_3[i] << ")" << endl;

else

cout << "Test Case #" << i+1 << " is Failed... (Input: " << input\_9\_3[i] << ", Output: " << output\_9\_3[i] << ")" << endl;

}

cin.clear();

cin.get();

cin.get();

return 0;

}

**Додаток Е**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | Test-Suite9.4 |
| Назва проекта / ПЗ  Name of Project / Software | Vasylenko-task.exe |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Vasylenko Dmytro |
| Виконавець  Implementer | Vasylenko Dmytro |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TC-1 | 1. Запустити застосунок. | Мова інтерфейсу – українська, у вікні застосунку наявні інструкції щодо роботи з програмою. | passed |
| TC-2 | 1. Запустити застосунок.  2. Ввести h | Вивід звукового сигналу.  Вікно застосунку:  " Для виходу з програми введiть v, або V, або A, iнакше повториться виконання програми." | passed |
| TC-3 | 1. Запустити застосунок.  2. Ввести j.  3. Ввести 9.  4. Ввести 25.  5. Ввести 1. | Вікно застосунку:  х у шістнадцятковій системі числення: 9  y у шістнадцятковій системі числення: 19  z у шістнадцятковій системі числення: 1  S = 4.99907" | passed |
| TC-4 | 1. Запустити застосунок.  2. Ввести j.  3. Ввести 9.  4. Ввести 25.  5. Ввести 1.  6. Ввести v. | Закриття вікна застосунку. | passed |
| TC-5 | 1. Запустити застосунок.  2. Ввести z.  3. Ввести 550. | Вікно застосунку:  "Кiлькiсть нарахованих бонусiв: 10  Сума до сплати: 547.5 грн." | passed |
| TC-6 | 1. Запустити застосунок.  2. Ввести x.  3. Ввести 250. | Вікно застосунку:  "Конвертованi градуси у шкалу Цельсiя: 121.111." | passed |
| TC-7 | 1. Запустити застосунок.  2. Ввести c.  3. Ввести 567. | Вікно застосунку:  "Кiлькiсть двiйкових одиниць у числi 567: 6" | passed |
| TC-8 | 1. Запустити застосунок.  2. Ввести c.  3. Ввести 567.  4. Ввести a. | Вікно застосунку:  " Введiть символ (латиницею):  j - для виклику функцiї s\_calсulation;  z - щоб дiзнатись кiлькiсть нарахованих бонусiв та суму до сплати (функцiя до задачi 9.1);  x - щоб конвертувати градуси за шкалою Фаренгейта у шкалу Цельсiя (функцiя до задачi 9.2);  с - щоб дiзнатись кiлькiсть двійкових одиниць або нулiв у числi (функцiя до задачi 9.3);" | passed |

**Додаток Є**

#include <iostream>

#include "ModulesVasylenko.h"

#include <limits>

#include <locale.h>

using namespace std;

void j\_s\_calculation()

{

int x, y, z;

cout << "Введiть число х: ";

cin >> x;

cout << "х у шiстнадцятковiй системi числення: " << hex << x << dec << endl;

cout << "Введiть число у: ";

cin >> y;

cout << "y у шiстнадцятковiй системi числення: " << hex << y << dec << endl;

cout << "Введiть число z: ";

cin >> z;

cout << "z у шiстнадцятковiй системi числення: " << hex << z << dec << endl;

cout << "S = " << s\_calculation(x,y, z);

}

void z\_task\_9\_1()

{

double price;

cout << "Введiть суму покупки у гривнях: ";

cin >> price;

cout << "Кiлькiсть нарахованих бонусiв: " << Bonus9\_1(price).bonus << "\nСума до сплати: " << Bonus9\_1(price).pay << " грн." << endl;

}

void x\_task\_9\_2()

{

float t;

cout << "Введiть кiлькiсть градусiв за шкалою Фаренгейта: ";

cin >> t;

cout << "Конвертованi градуси у шкалу Цельсiя: " << Temperature9\_2(t) << endl;

}

void c\_task\_9\_3()

{

int n;

cout << "Введiть цiле число від 0 до 51950: ";

do

{

cin >> n;

if (n>51950 || n<0)

cout << "Некоректно введенi данi, число повинно бути цiлим та від 0 до 51950. Спробуйте ще раз!" << endl;

}while (n>51950 || n<0);

if((n>>4)&1)

cout <<"Кiлькiсть двiйкових одиниць у числi "<< n << ": " << Binary9\_3(n) << endl;

else

cout <<"Кiлькiсть двiйкових нулiв у числi "<< n << ": " << Binary9\_3(n) << endl;

}

int main()

{

char \*locale=setlocale(LC\_ALL, "");

cout << "Виконав студент групи КІ-21-1 Василенко Д.М.\n"

"--------------------------------------------\n" << endl;

char symbol;

char command;

do

{

cout << "\nВведiть символ (латиницею):\n"

"j - для виклику функцiї s\_calculation;\n"

"z - щоб дiзнатись кiлькiсть нарахованих бонусiв та суму до сплати (функцiя до задачi 9.1);\n"

"x - щоб конвертувати градуси за шкалою Фаренгейта у шкалу Цельсiя (функцiя до задачi 9.2);\n"

"с - щоб дiзнатись кiлькiсть двійкових одиниць або нулiв у числi (функцiя до задачi 9.3);\n" << endl;

cin >> command;

switch (command)

{

case 'j':

j\_s\_calculation();

break;

case 'z':

z\_task\_9\_1();

break;

case 'х':

case 'x':

x\_task\_9\_2();

break;

case 'c':

case 'с':

c\_task\_9\_3();

break;

default: cout << "\a";

}

cout << "\nДля виходу з програми введiть v, або V, або A, iнакше повториться виконання програми." << endl;

cin >> symbol;

}while (!(symbol == 'v' || symbol == 'V' || symbol == 'A'));

return 0;

}