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

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КІ-20

Андрусик Б.М.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Поліщук Л. І.

Кропивницький – 2021

**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.

**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)"

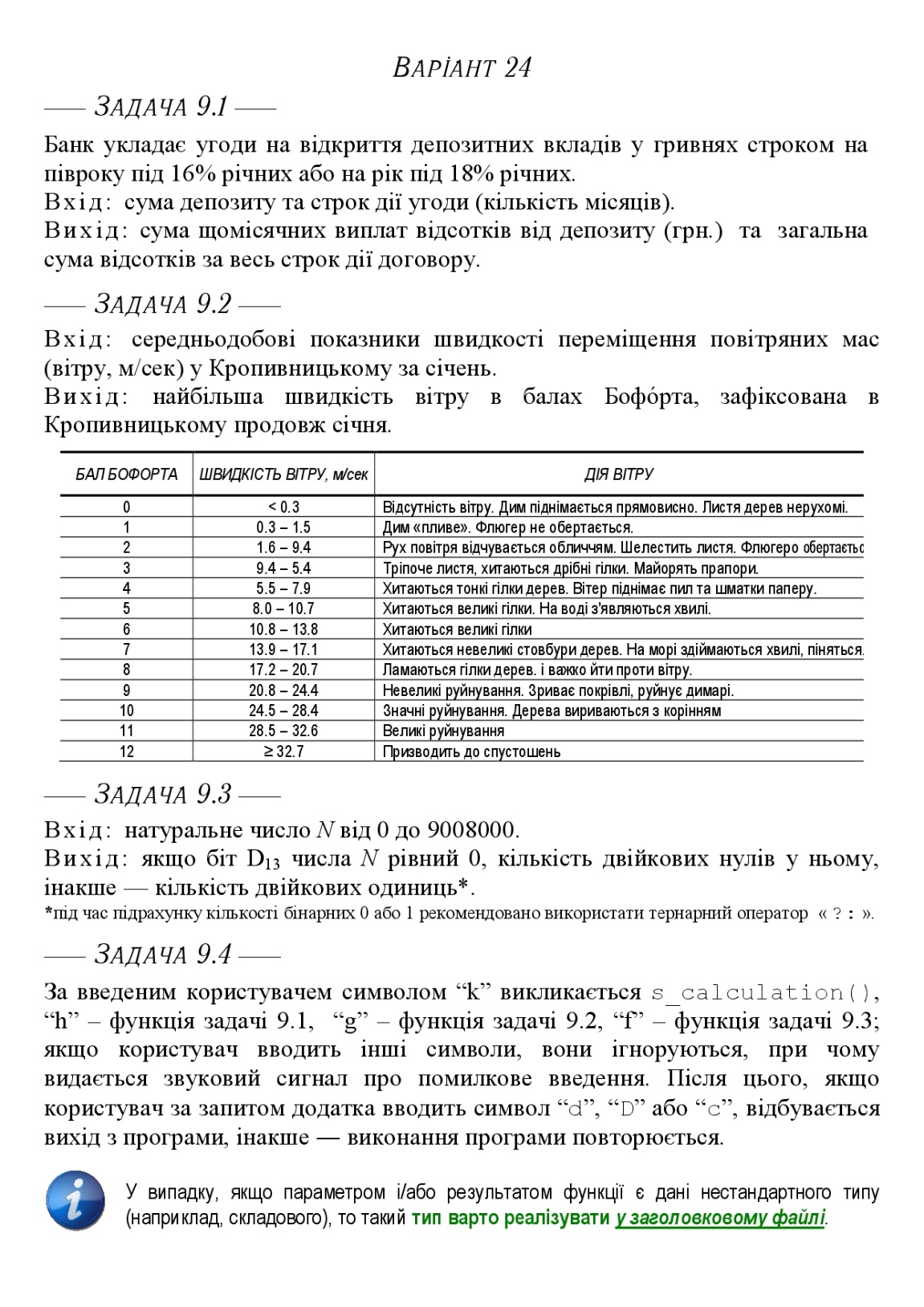
**Варіант: 24**

Рисунок 1 Завдання варіанту 10

**Завдання 9.1**

Постановка задачі:

ПОЧАТОК

1) Ввести суму депозиту

2) Ввести строк дії угоди

3) Вивести суму щомісячних виплат відсотків від депозиту

4) Вивести загальну суму відсотків за весь строк дії договору

КІНЕЦЬ

**Завдання 9.2**

Постановка задачі:

ПОЧАТОК

1) Ввести середньодобові показники швидкості переміщення повітряних мас

2) Вивести найбільшу швидкість вітру в балах Бофорта

КІНЕЦЬ

**Завдання 9.3**

Постановка задачі:

ПОЧАТОК

1) Ввести натуральне число від 0 до 9008000

2) Вивести кількість двійкових нулів або кількість двійкових одиниць

КІНЕЦЬ

Висновки:

Виконуючи дану лабораторну роботу було ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

В даній лабораторній роботі надається 4 різних задачі, виконуючи які вдалося використати різні оператори. Одним із перших завдань лабораторної роботи було заповнити два файли README, у яких було записано тему, варіант, мету й завдання до лабораторної роботи та створити теки: prj, Software, TestSuite, Report. Ці дані було завантажено до Git-репозиторію . Після цього було здійснено аналіз і постановку задач 9.1, 9,2 та 9,3,одержані дані було додано до звіту. Згодом було розроблено три набора контрольних прикладів до цих задач задля виконання модульного тестування, усі отримані тест-кейси були додані до \lab08\TestSuite та згодом додані до звіту як додаток.

Використовуючи вже створений під час виконання 8 лабораторної роботи проект статичної бібліотеки було створено 3 модуля. Для першої та третьої задачі це не склало жодних проблем невідмінно від другої задачі, де потрібно було отримувати тип даних int, а повертати дані типу char. Реалізація даного модулю зайняла найбільше часу. Оголосивши всі функції в заголовковому, що знаходиться в текі prj було створено проект консольного додатку “TestDriver”. За допомогою цього консольного додатку було виконано модульне тестування за допомогою тест-сьюта, що був створений раніше. Модульне тестування було виконане для фунцій отримало всі позначки passed.

Результати тест-кейсів повністю задовольняють очікувані результати. За допомогою тестування впевнився, що програмне забезпечення працює правильно. Усі результати тестування були задокументовані та додані до звіту. Вихідний код проектів та TestDriver включено до звіту як додаток. Ці дані було завантажено до Git-репозиторію.

Наступний крок – це здійснення аналіз і постановку задачі 9.4, що є маніпулятором для створених раніше функцій та функції з 8 лабораторної роботи s\_calculation. Після постановки задачі 9.4 був розроблений тест-сьют для виконання системного тестування програмного застосунку, який складається з тринадцяти тестових випадків, що є достатнім для тестування даної програми, файл тестового набору було збережено та усі отримані тест-кейси були додані до \lab09\TestSuite та згодом додані до звіту як додаток.

Завдяки математичної моделі та алгоритм описаний вербальним способом було реалізовано консольний застосунок мовою С++. Для коректної роботи та програми навігації програми по функціям було використано декілька змінних типу char. Також була створена валідація до кожної задачі, щоб відсіяти як умога більше не валідних даних. В результаті було отримано робочий застосунок. Всі тест-кейси пройшли тестування та отримали статус passed. Результати тест-кейсів повністю задовольняють очікувані результати. За допомогою тестування впевнився, що програмне забезпечення працює правильно. Файл тестового набору було збережено та усі отримані тест-кейси були додані до \lab09\TestSuite та згодом додані до звіту як додаток.

Завдяки даній лабораторній роботі було закріплено на практиці реалізації технології модульного програмування, застосування операторів С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

Додаток А

#include <iostream>

#include <math.h>

#include <string>

#include <clocale>

#include <stdlib.h>

#include <cmath>

#include <bitset>

#include <windows.h>

#include "ModulesAndrusyk.h"

using namespace std;

float s\_calculation(float x, float y, float z){

const double P = 3.141592653589793;

float S = fabs((pow((y\*z),fabs(x)))-(y/P)-(sqrt(x)));

return S;

}

void deposit()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

float annualRate = 0;

float depositSum = 0;

float profit = 0;

int monthAmount = 0;

cout << "Сума депозиту: ";

cin >> depositSum;

while (true)

{

if (depositSum <= 0)

{

cout << "Помилка! Введіть валідні дані: ";

cin >> depositSum;

}

else

{

break;

}

}

cout << "Вкажіть кількість місяців (6 або 12): ";

cin >> monthAmount;

while(true)

{

if(monthAmount == 6 || monthAmount == 12)

{

break;

}

else

{

cout << "Помилка! Введіть валідні дані: " ;

cin >> monthAmount;

}

}

switch (monthAmount)

{

case 6: annualRate = 16;

break;

case 12: annualRate = 18;

break;

}

cout << "Сума за місяць: ";

profit = (depositSum \* (annualRate / 100) / 12 \* monthAmount) / monthAmount;

cout << profit << endl;

cout << "Сума за весь час: ";

profit = depositSum \* (annualRate / 100) / 12 \* monthAmount;

cout << profit << endl;

}

void weather()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int SpeedBeaufs[31];

float SpeedMeters[31];

for (int i = 0; i < 31; i++)

{

SpeedBeaufs[i] = 0;

}

for (int i = 0; i < 31; i++)

{

cout << "Введіть показники за " << i + 1 << " число: ";

cin >> SpeedMeters[i];

}

for (int i = 0; i < 31; i++)

{

if(SpeedMeters[i] < 0.3)

{

SpeedBeaufs[i] = 0;

}

if(SpeedMeters[i] >= 0.3 && SpeedMeters[i] <= 1.5)

{

SpeedBeaufs[i] = 1;

}

if(SpeedMeters[i] >= 1.6 && SpeedMeters[i] <= 3.3)

{

SpeedBeaufs[i] = 2;

}

if(SpeedMeters[i] >= 3.4 && SpeedMeters[i] <= 5.4)

{

SpeedBeaufs[i] = 3;

}

if(SpeedMeters[i] >= 5.5 && SpeedMeters[i] <= 7.9)

{

SpeedBeaufs[i] = 4;

}

if(SpeedMeters[i] >= 8.0 && SpeedMeters[i] <= 10.7)

{

SpeedBeaufs[i] = 5;

}

if(SpeedMeters[i] >= 10.8 && SpeedMeters[i] <= 13.8)

{

SpeedBeaufs[i] = 6;

}

if(SpeedMeters[i] >= 13.9 && SpeedMeters[i] <= 17.1)

{

SpeedBeaufs[i] = 7;

}

if(SpeedMeters[i] >= 17.2 && SpeedMeters[i] <= 20.7)

{

SpeedBeaufs[i] = 8;

}

if(SpeedMeters[i] >= 20.8 && SpeedMeters[i] <= 24.4)

{

SpeedBeaufs[i] = 9;

}

if(SpeedMeters[i] >= 24.5 && SpeedMeters[i] <= 28.4)

{

SpeedBeaufs[i] = 10;

}

if(SpeedMeters[i] >= 28.5 && SpeedMeters[i] <= 32.6)

{

SpeedBeaufs[i] = 11;

}

if(SpeedMeters[i] > 32.6)

{

SpeedBeaufs[i] = 12;

}

}

int maxValue = SpeedBeaufs[0];

for (int i = 0; i < 31; i++)

{

if (SpeedBeaufs[i] > maxValue)

{

maxValue = SpeedBeaufs[i];

}

}

cout << endl << maxValue << " Бофортів" << endl;

}

void binaryNum()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string numBinary = "";

int numDecimal = 0;

int binZero = 0;

while (true)

{

cout << "Введіть число від 0 до 90080000: " << endl;

cin >> numDecimal;

if (numDecimal >= 0 && numDecimal < 90080000)

{

break;

}

else

{

cout << "Помилка! Ваше число виходить за вказані межі! Спробуйте ще раз!" << endl;

}

}

numBinary = bitset<16>(numDecimal).to\_string();

cout << numBinary << endl;

int i = 0;

while (numBinary[i])

{

if (numBinary[i] == '0') binZero++;

i++;

}

cout << " ^13 біт!" << endl;

// cout << "13 BIT: " << numBinary[numBinary.length() - 13] << endl;

if (numBinary[12] == '0')

{

cout << "Результат: " << binZero << endl;

}

else

{

cout << "Результат: " << 16 - binZero << endl;

}

}

Додаток Б

#ifndef MODULESANDRUSYK\_H\_INCLUDED

#define MODULESANDRUSYK\_H\_INCLUDED

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

void deposit();

void weather();

void binaryNum();

#endif // MODULESANDRUSYK\_H\_INCLUDED

Додаток В

#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

#include <locale.h>

#include <bitset>

#include <math.h>

#include "ModulesAndrusyk.h"

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << ("|Andrusyk Bohdan, CNTU|")<<endl;

char checkbox;

while (true)

{

cout << "Оберіть дію: ";

cin >> checkbox;

if (checkbox == 'k')

{

float a = 0, b = 0, c = 0;

cout << "x: ";

cin >> a;

cout << "y: ";

cin >> b;

cout << "z: ";

cin >> c;

cout << "Результат: " << s\_calculation(a, b, c) << endl;

}

if(checkbox == 'h')

{

deposit();

}

if(checkbox == 'g')

{

weather();

}

if(checkbox == 'f')

{

binaryNum();

}

if(checkbox == 'c' || checkbox == 'D' || checkbox == 'd')

{

break;

}

else{

Beep(500,500);

}

}

return 0;

}