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 КН-23

Горбенко К.О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

**Лабораторна робота №9**

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

*Мета***:** 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

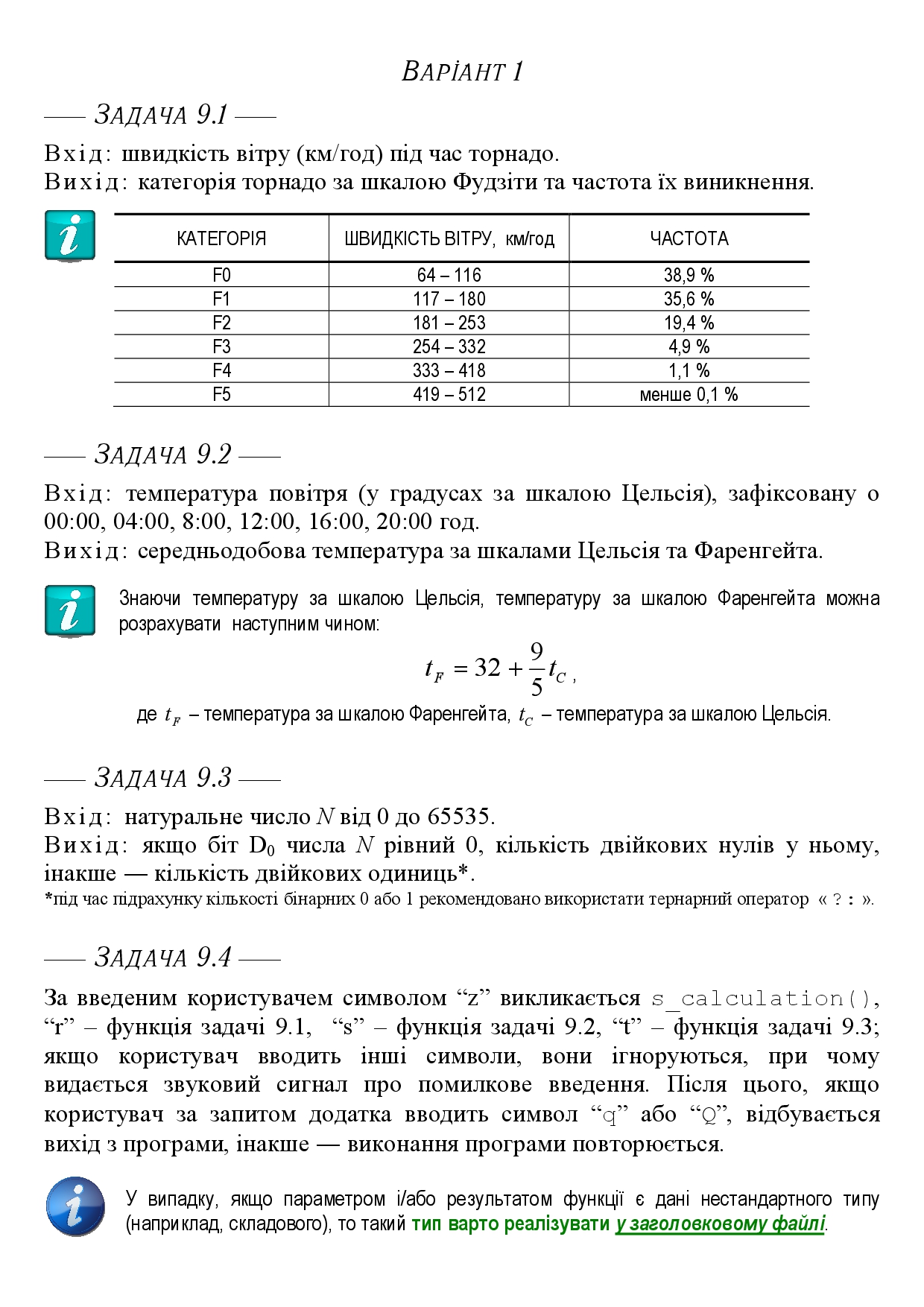
бібліотек, заголовкових файлів та програмних засобів у

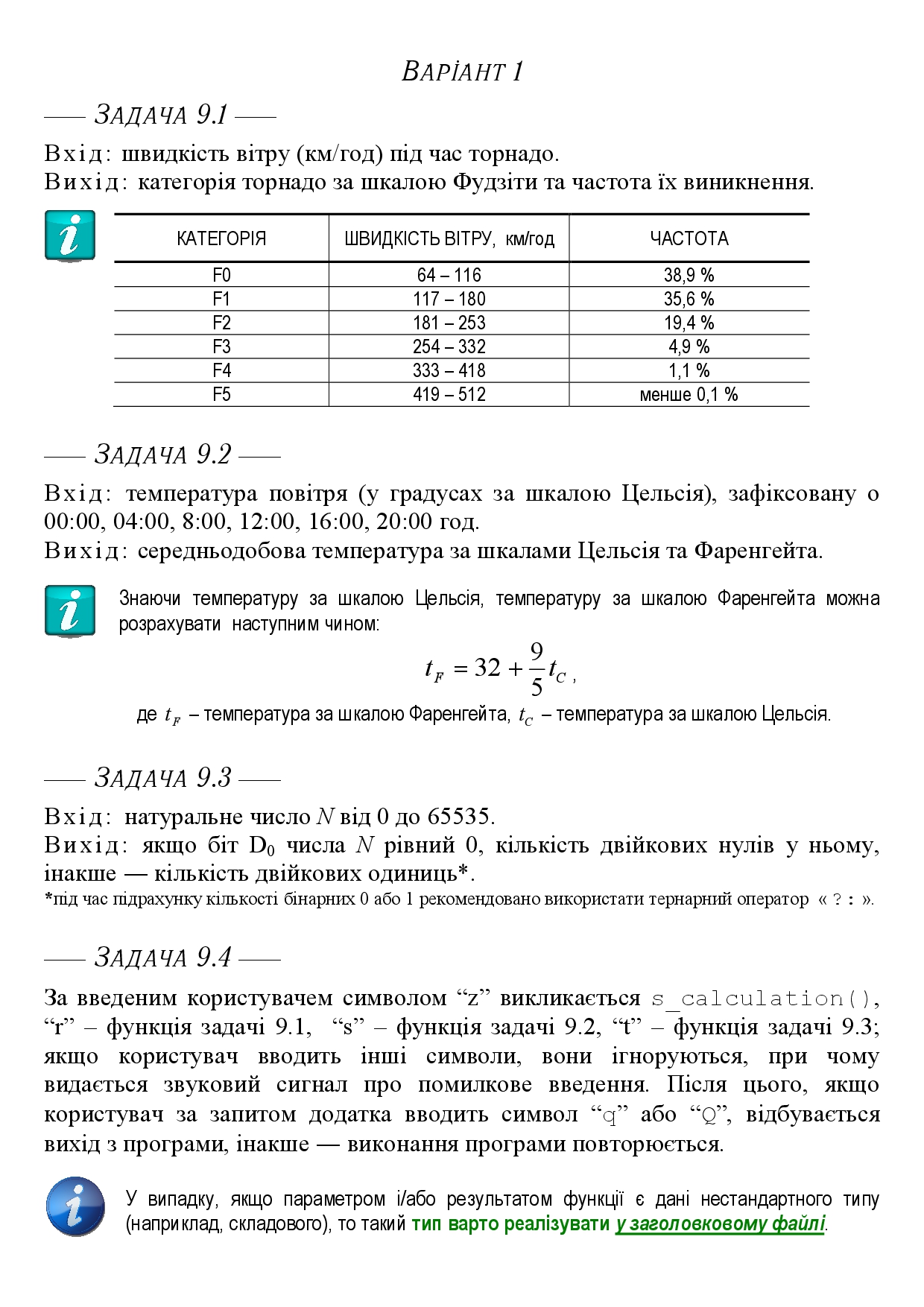
кросплатформовому середовищі Code::Blocks.

**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 1**



***Завдання 9.1***

*Вхідні дані:* speed – Дані типу int – Швидкість вітру

*Вихідні:* category – Дані типу string – Категорія та частота

*Якщо:* 64 – 116 , то категорія F0, *якщо:* 117 – 180 , то категорія F1, *якщо:* 181 – 253 , то категорія F2, *якщо:* 254 – 332 , то категорія F3, *якщо:* 333 – 418 , то категорія F4, *якщо:* 419 – 512 , то категорія F5, *інакше:* “Веденно не коректне значення. Ведіть число від 64 до 512”

***Завдання 9.2***

*Вхідні дані*: temp1-6 – Дані типу float – Температура за шість днів.

*Вихідні:* array[2] – Масив с двох чисел. Перше число Цельсія друге Фаренгейт.

*Обчислення велися за формулою*:

***Завдання 9.3***

*Вхідні дані:* N – Дані типу int – натуральне число від 0 до 65535

*Вихідні дані:* count – Дані типу int – кількість двійкових нулів або одиниць (залежить від числа N)

*Алгоритм*

1. Початок
2. Введення двух зміних count і first\_bit
3. Поки first\_bit менше або дорівнює числу num(числу):
4. Перевірка num, якщо = 0, то count + 1
5. Зсув вліво, перехід до наступного біта
6. Підрахунок кількості бінарних нулів або одиниць в залежності від значення найменш значущого біту
7. Кінець

***Лістинг ModulesHorbenko***

#include <cmath>

#include <iostream>

#include <string>

//=======================LAB8.1=======================

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

{

if (x < 2)

{

std::cout << "'Nan' Помилка, значення X менше 2 " << std::endl;

return 0;

}else

{

return pow(3 \* sin(sqrt(12 \* x + log10(x - 3))), y) + (z / x);

}

}

//=======================LAB9.1=======================

std::string tornado\_category(int speed)

{

std::string category;

if (speed >= 64 && speed <= 116) {

category = "Категорія торнадо: FO\nЧастота: 38,9%";

} else if (speed >= 117 && speed <= 180) {

category = "Категорія торнадо: F1\nЧастота: 35,6%";

} else if (speed >= 181 && speed <= 253) {

category = "Категорія торнадо: F2\nЧастота: 19,4%";

} else if (speed >= 254 && speed <= 332) {

category = "Категорія торнадо: F3\nЧастота: 4,9%";

} else if (speed >= 333 && speed <= 418) {

category = "Категорія торнадо: F4\nЧастота: 1,1%";

} else if (speed >= 419 && speed <= 512) {

category = "Категорія торнадо: F5\nЧастота: менше 0,1%";

} else {

category = "Веденно не коректне значення. Ведіть число від 64 до 512";

}

return category;

}

//=======================LAB9.2=======================

float\* calculate\_temp(float temp1, float temp2, float temp3, float temp4, float temp5, float temp6)

{

static float array[2];

float sum = temp1 + temp2 + temp3 + temp4 + temp5 + temp6;

float average\_celsius = sum / 6;

float average\_fahrenheit = (average\_celsius \* 9 / 5) + 32;

array[0] = average\_celsius;

array[1] = average\_fahrenheit;

return array;

}

//=======================LAB9.3=======================

unsigned int Bits(unsigned int num) {

int count = 0;

int first\_bit = 1;

while (first\_bit <= num) {

if ((num & first\_bit) == 0) {

count++;

}

first\_bit <<= 1;

}

return (num >> 15) ? (16 - count) : count;

}

***Лістинг TestDriver***

#include <iostream>

#include <string>

#include <conio.h>

#include "ModulesHorbenko.h"

using namespace std;

void testTornadoCategory(int test\_num, int speed, const string& expected)

{

string result = tornado\_category(speed);

cout << "=============Test\_9.1==============" << endl;

if (result == expected)

{

cout << "Test\_" << test\_num << "\tPassed" << endl;

cout << "Result: " << result << endl;

}

else

{

cout << "Test\_" << test\_num << "\tFailed" << endl;

}

}

void testCalculateTemp(int ts\_num, float temp1, float temp2, float temp3, float temp4, float temp5, float temp6, float cel, float fahr)

{

float \*array = calculate\_temp(temp1, temp2, temp3, temp4, temp5, temp6);

cout << "=============Test\_9.2==============" << endl;

if (array[0] == cel || array[1] == fahr)

{

cout << "Test\_" << ts\_num << "\tPassed" << endl;

cout << "Очікований результат:\n" << "Цельсія: " << cel << "\nФаренгейт: " << fahr << endl;

cout << "\nРезультат функції:" << endl;

cout << "Цельсія: " << array[0] << endl;

cout << "Фаренгейт: " << array[1] << endl;

}else

{

cout << "Test\_" << ts\_num << "\tFailed" << endl;

cout << "Очікований результат:\n" << "Цельсія: " << cel << "\nФаренгейт: " << fahr << endl;

float \*array = calculate\_temp(temp1, temp2, temp3, temp4, temp5, temp6);

cout << "\nРезультат функції:" << endl;

cout << "Цельсія: " << array[0] << endl;

cout << "Фаренгейт: " << array[1] << endl;

}

}

void testCountBits(int ts\_num, unsigned int num, unsigned int expected)

{

unsigned int result = Bits(num);

cout << "=============Test\_9.3==============" << endl;

if (result == expected)

{

cout << "Test\_" << ts\_num << "\tPassed" << endl;

cout << "Result: " << result << endl;

} else

{

cout << "Test\_" << ts\_num << "\tFailed" << endl;

}

}

int main()

{

system("chcp 1251 & cls");

// tornado\_category

testTornadoCategory(1, 323, "Категорія торнадо: F3\nЧастота: 4,9%");

testTornadoCategory(2, 404, "Категорія торнадо: F4\nЧастота: 1,1%");

testTornadoCategory(3, 600, "Веденно не коректне значення. Ведіть число від 64 до 512");

// calculate\_temp

testCalculateTemp(1, 10, 20, 30, 25, 20, 15, 20, 68);

testCalculateTemp(2, -5, 0, 5, 10, 15, 20, 7.5, 45.5);

testCalculateTemp(3, 12.5, 10.5, 18.7, 20.3, 25.6, 30.2, 19.6333, 67.34);

// Bits

testCountBits(1, 4096, 12);

testCountBits(2, 12, 2);

testCountBits(3, 65535, 16);

getch();

}  
  
***Результати виконання TestDriver***

=============Test\_9.1==============

Test\_1 Passed

Result: Категорія торнадо: F3

Частота: 4,9%

=============Test\_9.1==============

Test\_2 Passed

Result: Категорія торнадо: F4

Частота: 1,1%

=============Test\_9.1==============

Test\_3 Passed

Result: Веденно не коректне значення. Ведіть число від 64 до 512

=============Test\_9.2==============

Test\_1 Passed

Очікований результат:

Цельсія: 20

Фаренгейт: 68

Результат функції:

Цельсія: 20

Фаренгейт: 68

=============Test\_9.2==============

Test\_2 Passed

Очікований результат:

Цельсія: 7.5

Фаренгейт: 45.5

Результат функції:

Цельсія: 7.5

Фаренгейт: 45.5

=============Test\_9.2==============

Test\_3 Passed

Очікований результат:

Цельсія: 19.6333

Фаренгейт: 67.34

Результат функції:

Цельсія: 19.6333

Фаренгейт: 67.34

=============Test\_9.3==============

Test\_1 Passed

Result: 12

=============Test\_9.3==============

Test\_2 Passed

Result: 2

=============Test\_9.3==============

Test\_3 Passed

Result: 16

***Задача 9.4***

*Вхідні дані:* input – Дані типу char – викликає потрібну функцію за літерою с модуля

*Вихідні дані:* Виведення результату певної функції, яку обрав користувач

В коді використовується switch-case для вибору функції згідно з введеним користувачем символом( 'z', 'г', 's', 't', 'q' або 'Q'), після чого запитується ввід параметрів для цієї функції та виводиться результат виклику функції.

Якщо користувач вводить невірний символ, програма видає звук, який оповіщає про помилку та пропонує повторити введення.

***Лістинг проекту Horbenko\_task:***

#include <iostream>

#include <conio.h>

#include "ModulesHorbenko.h"

using namespace std;

int main()

{

system("chcp 1251 & cls");

char input;

float x, y, z;

int speed;

float temp1, temp2, temp3, temp4, temp5, temp6;

while (true) {

cout << "Введіть символ ('z', 'г', 's', 't', 'q' або 'Q'): ";

cin >> input;

switch (input) {

case 'z':

cout << "Введіть значення x, y та z через пробіл: ";

cin >> x >> y >> z;

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

break;

case 'r':

cout << "Введіть швидкість торнадо: ";

cin >> speed;

cout << tornado\_category(speed) << endl;

break;

case 's':

cout << "Введіть шість температур через пробіл\n00:00, 04:00, 8:00, 12:00, 16:00, 20:00 год\n ";

cin >> temp1 >> temp2 >> temp3 >> temp4 >> temp5 >> temp6;

float\* result;

result = calculate\_temp(temp1, temp2, temp3, temp4, temp5, temp6);

cout << "Середня температура у Цельсіях: " << result[0] << endl;

cout << "Середня температура у Фаренгейтах: " << result[1] << endl;

break;

case 't':

unsigned int num;

cout << "Введіть ціле число: ";

cin >> num;

cout << "Кількість нульових бітів у бінарному представленні числа: " << Bits(num) << endl;

break;

case 'q':

case 'Q':

cout << "Вихід з програми." << endl;

exit(0);

default:

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

cout << '\a';

}

}

getch();

return 0;

}

***Контрольні запитання і завдання:***

1. *Яке призначення операцій інкремента та декремента?В чому полягає відмінність між їх префіксною і постфіксною формами у мові програмування С/С++?*

Операції інкремента (++) та декремента (--) використовуються для збільшення або зменшення значення змінної на 1 відповідно. Префіксна форма (++x, --x) спочатку змінює значення змінної, а потім використовує його в виразі. Постфіксна форма (x++, x--) використовує поточне значення змінної в виразі, а потім змінює його.

1. *Яке значення матиме mch при char mch = 'D'; --mch; ? Поясніть отриманий результат.*

Змінна mch матиме значення 'C'. Постфіксний декремент --mch спочатку використовує значення mch ('D') в виразі, а потім зменшує його на 1, отримуючи 'C'.

1. *Якими є призначення й синтаксис запису складових операторів присвоювання в мові програмування С/С++?*

Оператори присвоювання (=, +=, -=, \*=, /=, %=) використовуються для присвоєння значень змінним. Синтаксис: змінна = вираз;

1. *Перелічіть оператори логічних операцій С/С++, типи їх операндів і результату.*

Оператори логічних операцій в С/С++: && (логічне І), || (логічне АБО), ! (логічне НЕ). Операнди - булеві значення (true або false), результат - також булеве значення.

1. *Яким буде значення comp при bool comp = (!0 && 1) == 0; ?Отриманий результат обґрунтуйте.*

Значення comp буде false. Вираз (!0 && 1) розглядається як (true && true), що дає true, але результат порівняння з 0 (true == 0) дає false, оскільки false не рівне 0.

1. *Перелічіть оператори порозрядних операцій С/С++, типи їх операндів і результату виконання.*

Оператори порозрядних операцій в С/С++:

* & (порозрядне І),
* | (порозрядне АБО),
* ^ (порозрядне виключне АБО),
* ~ (порозрядне НЕ),
* << (порозрядний зсув вліво),
* >> (порозрядний зсув вправо).

Типи операндів - цілі числа (int, long, short), результат - також ціле число.

1. *Розташуйте в порядку спадання пріоритету виконання наступних операцій: логічні, арифметичні, порівняння, інкремент/декремент, складові операції присвоювання.*

- Інкремент/декремент (++, --)

- Арифметичні (\*, /, %, +, -)

- Порівняння (==, !=, <, >, <=, >=)

- Логічні (&&, ||)

- Складові операції присвоювання (=, +=, -=, \*=, /=, %=)

1. *Що розуміють під асоціативністю операторів мови програмування? Наведіть приклад.*

Асоціативність операторів визначає порядок виконання операцій з однаковим пріоритетом. Наприклад, у виразі a + b + c асоціативність оператора + вказує на те, що спочатку виконується a + b, а потім результат додається до c.

1. *Яке значення матиме n при: int n = 0; n = !n<<1&1; ?*

Значення буде 0 (false)

1. *Якими є призначення і синтаксис запису операторів повного й неповного розгалуження С/С++?*

// Повне розгалуження

if (умова) {

// інструкції, якщо умова істинна

} else {

// інструкції, якщо умова хибна

}

// Неповне розгалуження

if (умова) {

// інструкції, якщо умова істинна

}

1. *Сформулюйте алгоритм виконання тернарного оператора (операції умови) мови програмування С/С++.*

Алгоритм виконання тернарного оператора (операції умови) у мові програмування С/С++:

* Оцінити умову (вираз перед ?).
* Якщо умова true, виконати перший вираз після ?.
* Якщо умова false, виконати другий вираз після :.

1. *У чому полягає відмінність між результатом виконання оператора break та оператора continue? Перелічіть випадки, у яких є доцільним використання кожного з них під час реалізації програмного забезпечення.*

Оператор break припиняє виконання циклу або switch-блоку і переходить до наступного виконуваного рядка після циклу або switch. Оператор continue перериває поточну ітерацію циклу і переходить до наступної ітерації циклу.

Випадки, коли варто використовувати break:

* Коли необхідно припинити виконання циклу.
* При вибору необхідно вийти з циклу.

Випадки, коли варто використовувати continue:

* При необхідності перервати поточну ітерацію циклу і перейти до наступної ітерації.
* Коли деякі обчислення потрібно пропустити для певних умов.

1. *Яке призначення циклів у програмуванні та які розрізняють їх види у мові програмування С/С++?*

Цикли в програмуванні використовуються для повторення виконання певного блоку коду доти, доки задана умова виконання є true. У мові програмування С/С++ існують кілька видів циклів:

Цикл while: виконується, поки умова true.

Цикл do-while: виконується хоча б один раз, а потім поки умова true.

Цикл for: виконується зазначену кількість разів або до виконання умови.

1. *Сформулюйте алгоритм виконання оператора switch. Якого типу мають бути його перемикаючий і константні вирази?*

Алгоритм виконання оператора switch у мові С/С++:

Вираз у switch порівнюється з кожним константним виразом case.

Якщо знайдено відповідність, виконується код відповідного case, а потім виходить з switch.

Якщо не знайдено відповідності, виконується код в розділі default (якщо він є).

Перемикаючий вираз у switch повинен бути цілого типу (int або char) або enum, а константні вирази у case - цілі числа або char.

1. *Яким є синтаксис запису циклів for, while, do…while С/С++? Сформулюйте алгоритми їх виконання.*

// for

for (ініціалізація; умова; інкремент) {

// блок коду

}

// while

while (умова) {

// блок коду

}

// do…while

do {

// блок коду

} while (умова);

Алгоритми виконання:

- for ініціалізує змінну, перевіряє умову; якщо вона істинна, виконує блок коду, а потім інкрементує змінну.

- while перевіряє умову перед кожною ітерацією; якщо вона істинна, виконує блок коду.

- do…while виконує блок коду один раз, а потім перевіряє умову; якщо вона істинна, продовжує виконання.

1. *Яке призначення кваліфікатора типів const С/С++? Наведіть приклад і перелічіть випадки доцільності його застосування.*

Випадки доцільності застосування const:

* Захист від непередбачуваних змін у значенні змінних.
* Оптимізація коду, оскільки компілятор може здійснити деякі оптимізації, знаючи, що значення константи не змінюється.
* Чіткіше вираження інтенції програміста, що певна змінна має бути сталим.

Приклад:

const int x = 5;

x = 10; // Помилка: константу не можна змінювати

1. *Який з виразів оператора for (ініціалізації, умови, модифікації/ оновлення) є обов’язковим у С/С++? Відповідь обґрунтуйте*

У операторі for у C/C++ обов’язковим є лише вираз умови. Ініціалізація та модифікація/оновлення не є обов’язковими і можуть бути виключені, якщо це необхідно.

1. *Що буде виведено об’єктом cout при: short int b = 0; cout << hex << ( ~b&0x80 ); ?*

Вираз cout << hex << ( ~b&0x80 ); виведе 80.

1. *Сформулюйте алгоритм перетворення функції С/С++ у функцію статичної бібліотеки С/С++, а також спосіб її використання під час реалізації програмного забезпечення.*

Для перетворення функції C/C++ у функцію статичної бібліотеки, потрібно:

1. Створити статичну бібліотеку

2. Додати код функції у вихідний код бібліотеки

3. Додати прототип функції у окремий заголовковий файл

4. Скомпілювати статичну бібліотеку у об’єктний файл

5. Підключити статичну бібліотеку до програми за допомогою лінкера

6. Підключити заголовковий файл статичної бібліотеки до програми

1. *Наведіть приклад опису константи у заголовковому файлі С/С++ та її використання у програмі.*

У заголовковому файлі (constants.h):

#ifndef CONSTANTS\_H // Захист від повторного включення

#define CONSTANTS\_H

const double PI = 3.14159; // Оголошення константи PI

#endif

У програмі (main.cpp):

#include <iostream>

#include "constants.h" // Підключення заголовкового файлу з константами

int main() {

std::cout << "Value of PI: " << PI << std::endl; // Використання константи PI

return 0;

}

У цьому прикладі, константа PI оголошується у заголовковому файлі constants.h і підключається до програми main.cpp за допомогою директиви #include "constants.h". Підключення заголовкового файлу дозволяє використовувати цю константу у всіх файлах, які підключають цей заголовковий файл.