Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10

з навчальної дисципліни

“Базові методології та технології програмування”

Реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням

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

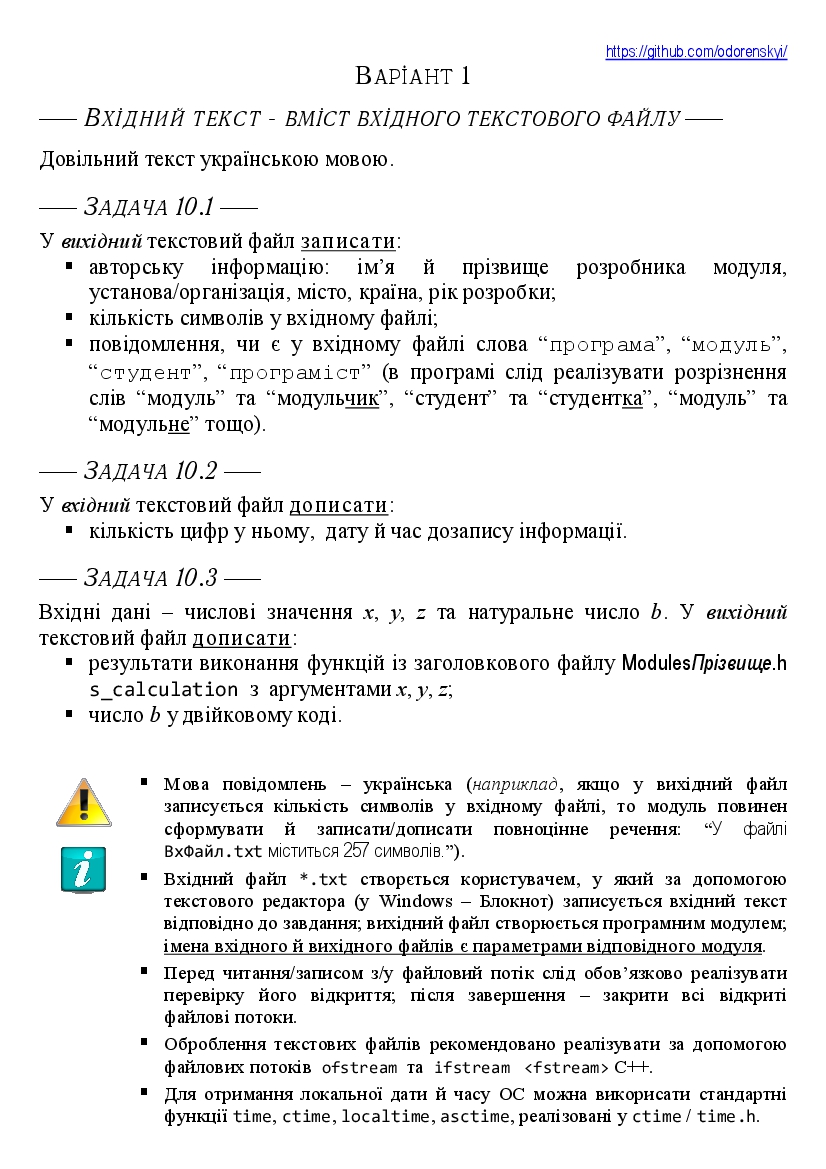
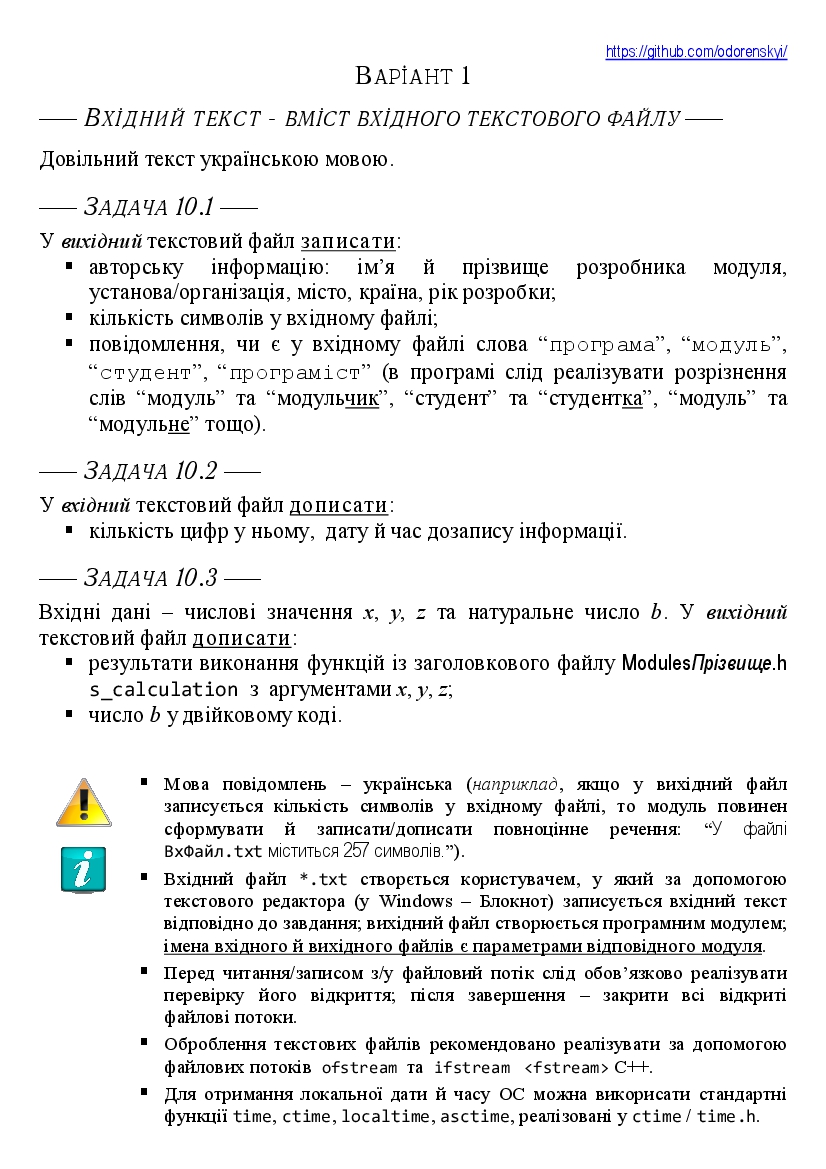
**Лабораторна робота №10**

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

*Мета***:** Набуття ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьної інформації.

**Завдання до лабораторної роботи:**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище лабораторних робіт №8–9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3



***Завдання 10.1***

*Вхідні дані:* inputFileName, outputFileName – вхідний та вихідний фали

*Вихідні дані:* вихідний текстовий файл в якому міститься авторська інформація, кількість символів файлу inputFileName та повідомленя про вміст слів у файлі (“програма””,”модуль”,”студент”, “програміст”)

*Опис:* Щоб могти працювати з файлами нам потрібно використовувати файлові потоки <fstream>. Ми використовуємо inputFile для вхідного файлу і outputFile для вихідного файлу . Після читання вмісту файлу у зміну content, ми обчислюється кількість символів у рядку content за допомогою методу size(). Далі створюються масиви words і wordFound, які містять слова для пошуку та прапорці їх наявності. В циклі слова з масиву words шукаються у рядку content. Якщо слово знайдено, відповідний прапорець у масиві wordFound встановлюється в true.

***Завдання 10.2***

*Вхідні дані*: filename – вхідний файл

*Вихідні дані:* у вхідний файл filename виконується до запис інформації про кількість чисел та про час запису інформації

*Опис:* По прикладу першого завдання ми знову використовуємо ofstream щоб відкрити файл для до запису, та ifstream для зчитування. Потім ми отримуємо поточний час за допомогою функцій time. Після в нас йде цикл який зчитує файл посимвольно та перевіряє, чи є зчитаний символ цифрою якщо є то збільшує лічильник на один.

***Завдання 10.3***

*Вхідні дані:* x, y, z – числові значення типу float, b – натуральне число типу unsigned int, outputFileName – вихідний файл

*Вихідні дані:* Результат виконання функції з ModulesHorbenko s\_calculation з аргументами x, y, z. Число b у двійковому коді.

*Опис:* Подібно до минулих завдань здійснюємо відкриття файлу для до запису. З ModuleHorbenko використовуємо функцію s\_calculation та виводимо результат у файл разом з аргументами. Після в нас запуусскається цикл for який починається з значущого біта і веде цикл до менш значущого, в циклі використовується маска яка отримає біт на позиції i. В кінці циклу відбувається до запис за допомогою тернарного оператора ‘1’ або ‘0’.

***Реалізація виконання TestDriver:***

=============Test\_10.1==============

Test\_1

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

Операція завершена. Результати записані у вихідний файл.

Перевірити виконання того що тест-сьюта та результата програми,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_2

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

Операція завершена. Результати записані у вихідний файл.

Перевірити виконання того що тест-сьюта та результата програми,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_3

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

Операція завершена. Результати записані у вихідний файл.

Перевірити виконання того що тест-сьюта та результата програми,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_4

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

Операція завершена. Результати записані у вихідний файл.

Перевірити виконання того що тест-сьюта та результата програми,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_5

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

Операція завершена. Результати записані у вихідний файл.

Перевірити виконання того що тест-сьюта та результата програми,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

=============Test\_10.2==============

Test\_1

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_2

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_3

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_4

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_5

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

=============Test\_10.3==============

Test\_1

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_2

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_3

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_4

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

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

Test\_5

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

Перевірте зміст файлу з тест-сьютом,

якщо воно сходиться натисніть 'y', якщо ні то іншу кнопку

Passed

***Лістинг ModulesHorbenko***

#include <cmath>

#include <iostream>

#include <string>

#include <fstream>

#include <ctime>

using namespace std;

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

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

{

if (x < 2)

{

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

return 0;

}else

{

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

}

}

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

string tornado\_category(int speed)

{

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;

}

//=======================LAB10.1=======================

void processfile(const string& inputFileName, const string& outputFileName) {

ifstream inputFile(inputFileName);

ofstream outputFile(outputFileName);

string content((istreambuf\_iterator<char>(inputFile)), istreambuf\_iterator<char>());

inputFile.close();

int charCount = content.size();

string words[] = {"програма", "модуль", "студент", "програміст"};

bool wordFound[] = {false, false, false, false};

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

size\_t pos = content.find(words[i]);

while (pos != string::npos) {

wordFound[i] = true;

pos = content.find(words[i], pos + 1);

}

}

outputFile << " ====================================" << endl;

outputFile << "||Автор: Горбенко Катерина ||" << endl;

outputFile << "||Установа: Центральноукраїнський ||" << endl;

outputFile << "||національний технічний університет||" << endl;

outputFile << "||Місто: Кропівницький ||" << endl;

outputFile << "||Країна: Україна ||" << endl;

outputFile << "||Рік розробки: 2024 ||" << endl;

outputFile << " ====================================" << endl;

outputFile << "Кількість символів у вхідному файлі: " << charCount << endl;

outputFile << "======================================" << endl;

outputFile << "Результати перевірки на наявність слів:" << endl;

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

outputFile << words[i] << ": " << (wordFound[i] ? "Знайдено" : "Не знайдено") << endl;

}

outputFile << "======================================" << endl;

outputFile.close();

cout << "Операція завершена. Результати записані у вихідний файл." << endl;

}

//=======================LAB10.2=======================

void appendfileInfo(const string& filename) {

ofstream file(filename, ios\_base::app);

time\_t currentTime = time(nullptr);

string dateTime = asctime(localtime(&currentTime));

ifstream inputFile(filename);

int digitCount = 0;

char ch;

while (inputFile.get(ch)) {

if (isdigit(ch)) {

digitCount++;

}

}

file << "Кількість цифр: " << digitCount << "\n";

file << "Дата та число: " << dateTime;

file.close();

}

//=======================LAB10.3=======================

void binary(float x, float y, float z, unsigned int b, const string& outputFileName) {

ofstream outputsFile(outputFileName, ios\_base::app);

outputsFile << "Результат функцій ModulesHorbenko.h:" << endl;

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

outputsFile << "Число у двійковому коді: " << b << ": ";

for (int i = 31; i >= 0; i--) {

unsigned int mask = 1 << i;

outputsFile << ((b & mask) ? '1' : '0');

}

outputsFile << endl;

outputsFile.close();

}

***Лістинг TestDriver***

#include <cmath>

#include <iostream>

#include <fstream>

#include <string>

#include <conio.h>

#include "ModulesHorbenko.h"

using namespace std;

void testprocessfile(int test\_num, const string& inputFileName, const string& outputFileName)

{

cout << "\nTest\_" << test\_num << endl;

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

processfile(inputFileName, outputFileName); // Використання переданих значень

cout << "Перевірити виконання того що тест-сьюта та результата програми, \nякщо воно сходиться натисніть 'y', якщо ні то іншу кнопку "<< endl;

char button = getch();

if (button == 'y')

{

cout << "Passed" << "\n====================================" << endl;

}else

{

cout << "Failed" << "\n====================================" << endl;

}

}

void testappendfileInfo(int test\_num, const string& filename)

{

cout << "\nTest\_" << test\_num << endl;

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

appendfileInfo(filename);

cout << "Перевірте зміст файлу з тест-сьютом, \nякщо воно сходиться натисніть 'y', якщо ні то іншу кнопку " << endl;

char button = getch();

if (button == 'y')

{

cout << "Passed" << "\n====================================" << endl;

}else

{

cout << "Failed" << "\n====================================" << endl;

}

}

void testbinary(int test\_num, float x, float y, float z, unsigned int b, const string& outputFileName)

{

cout << "\nTest\_" << test\_num << endl;

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

binary(x, y, z, b, outputFileName);

cout << "Перевірте зміст файлу з тест-сьютом, \nякщо воно сходиться натисніть 'y', якщо ні то іншу кнопку " << endl;

char button = getch();

if (button == 'y')

{

cout << "Passed" << "\n====================================" << endl;

}

else

{

cout << "Failed" << "\n====================================" << endl;

}

}

int main(){

system("chcp 1251 & cls");

cout << "=============Test\_10.1==============" << endl;

testprocessfile(1, "input\_1.txt", "output\_1.txt");

testprocessfile(2, "input\_2.txt", "output\_2.txt");

testprocessfile(3, "input\_3.txt", "output\_3.txt");

testprocessfile(4, "input\_4.txt", "output\_4.txt");

testprocessfile(5, "input\_5.txt", "output\_5.txt");

cout << "=============Test\_10.2==============" << endl;

testappendfileInfo(1, "input\_1.txt");

testappendfileInfo(2, "input\_2.txt");

testappendfileInfo(3, "input\_3.txt");

testappendfileInfo(4, "input\_4.txt");

testappendfileInfo(5, "input\_5.txt");

cout << "=============Test\_10.3==============" << endl;

testbinary(1, 5, 13, 1, 1, "binary\_1.txt");

testbinary(2, 5, 2, 0, 1567890, "binary\_2.txt");

testbinary(3, 10, 2, 5, 56, "binary\_3.txt");

testbinary(4, 50, 15, 100, 11, "binary\_4.txt");

testbinary(5, 3, 2, 5, 96, "binary\_5.txt");

return 0;

}

***Висновок:***

Провівши цю лабораторну роботу я набула практичних навичок у реалізації програмних модулів у середовищі Code::Blocks з використанням мови програмування C++. Завдання включали розгалужені та ітераційні обчислювальні процеси з використанням різних типів даних, оброблення файлів та роботу з файловими потоками.

Перше завдання передбачало створення програмного модуля, який аналізує вміст вхідного текстового файлу. Ключовими елементами були відкриття файлу для зчитування, обробка тексту для підрахунку кількості символів та перевірка наявності певних слів у тексті. Результати аналізу записувалися у вихідний текстовий файл.

Функція яка реалізує завдання 10.1 (processfile):

Ця функція використовується для аналізу вмісту текстового файлу та створення звіту про його вміст. Вхідні дані для цієї функції - це ім'я вхідного файлу inputFileName і ім'я вихідного файлу outputFileName. Функція відкриває вхідний файл для зчитування та створює рядок content, який містить весь текст з файлу. Далі виконується підрахунок кількості символів у файлі, а також перевіряється наявність певних слів, таких як "програма", "модуль", "студент", "програміст". Результати цих аналізів записуються у вихідний файл outputFileName, разом з авторською інформацією.

У другому завданні програмний модуль отримував вхідний файл і записував у нього інформацію про кількість чисел у файлі та час запису цієї інформації. Для цього використовувалися засоби роботи з файловими потоками та операції для обробки числових значень у файлі.

Функція яка реалізує завдання 10.2 (appendfileInfo):

Ця функція призначена для додавання інформації до вже існуючого файлу. Вхідні дані - це ім'я файлу filename, до якого потрібно додати інформацію. Функція відкриває файл для дозапису (ios\_base::app), отримує поточний час та додаткову інформацію про файл, наприклад, кількість цифр у файлі. Ця інформація додається до вихідного файлу.

Трете завдання включало обробку числових даних та виведення результату обчислення функції разом з аргументами у вихідний файл. Також в цьому завданні було виведення натурального числа у двійковому коді. Для цього використовувалися побітові операції та функція з попередніх завдань.

Функція яка реалізує завдання 10.3 (binary):

Ця функція виконує два основних завдання:

Вона обчислює результат виконання функції s\_calculation з аргументами x, y, z і записує цей результат у вихідний файл.

Вона виводить натуральне число b у двійковому форматі у вихідний файл. Для цього використовується побітове зчитування кожного біта числа b та запис його у вихідний файл у двійковому вигляді.

Тестувальний драйвер TestDriver:

Цей драйвер був створений для автоматизованого тестування програмних модулів. Він викликав різні тести для кожного з модулів і перевіряв правильність їх роботи. Після виконання кожного тесту він надавав звіт про результати, які можна було оцінити як "Passed" або "Failed".

Всі ці завдання разом дозволили набути практичних навичок у роботі з файлами, обробці тексту та числових даних, використанні побітових операцій та автоматизованому тестуванні програмних модулів. Такий комплексний підхід допоміг зрозуміти, як реалізувати різні аспекти програмування на мові C++.

***Контрольні запитання і завдання***

1. Поясніть сутність понять області видимості об’єктів та простору імен у мові програмування С++.

Область видимості об'єктів визначає, де ці об'єкти можуть бути доступні в програмі. Простір імен визначає, як імена об'єктів пов'язуються з їхніми знаходженнями в програмі.

2. Яким є час життя локального об’єкта (змінної)? Наведіть приклад оголошення локальної та глобальної змінних одного зі складових типів С/С++.

Час життя локального об'єкта - від моменту його створення до закінчення блоку, в якому він оголошений.

3. Якою є область видимості змінної, оголошеної у циклі С++? Відповідь обґрунтуйте та доведіть на прикладі.

Змінна, оголошена в циклі, має область видимості, обмежену цим циклом. Вона недоступна за межами циклу.

4. Якими є призначення й синтаксис запису директиви extern? Наведіть приклад її застосування під час реалізації міжмодульних змінних або функцій.

Директива extern вказує, що змінна або функція вже була оголошена в іншому модулі програми.

Приклад:

// В одному файлі

extern int globalVar;

// В іншому файлі

int globalVar = 10;

5. У чому полягає відмінність між масивом і рядком С/С++ з погляду реалізації їх оброблення?

Основна відмінність полягає в тому, що масив - це набір однотипних елементів, тоді як рядок - це масив символів, що завершується нульовим символом '\0'. Обробка рядків зазвичай включає в себе використання рядкових функцій, тоді як масиви можуть бути оброблені звичайними масивними операціями.

6. Яким є синтаксис запису та оголошення об’єднання у С/С++?

Об'єднання дозволяє зберігати різні типи даних в одній області пам'яті. В кожен момент часу в об'єднанні може зберігатися лише одне значення.

Синтаксис:

union MyUnion {

int i;

float f;

char str[20];

};

7. Дайте визначення винятку (exception). Як у С++ реалізовується його оброблення?

Виняток (exception) - це особлива ситуація, яка виникає під час виконання програми і порушує її нормальний хід. Виняток може бути викликаний різними причинами, такими як ділення на нуль, спроба звернутися до неіснуючого елемента масиву, нестача пам'яті тощо.

Приклад:

try {

// код, де можуть виникати винятки

int x = 5 / 0; // виникне виняток ділення на нуль

}

catch (const std::exception& e) {

// обробка винятку

std::cout << "Виняток: " << e.what() << std::endl;

}

8. Яким є синтаксис оголошення одно- й двовимірних масивів даних у С/С++ та за допомогою якої операції і за якими індексами здійснюється доступ до їх елементів?

1. Оголошення:

• Одновимірний масив: тип\_даних назва\_масиву[розмір];

• Двовимірний масив: тип\_даних назва\_масиву[розмір1][розмір2];

2. Доступ

• Доступ до елементів в одновимірний масив: назва\_масиву[індекс]

• Доступ до елементів в двовимірний масив: назва\_масиву[індекс1][індекс2]

9. Яким є призначення наступних рядкових функцій з заголовкового файлу cstring (string.h): strstr, strlen, strcpy, strncpy, strcat, strncat, strcmp, strncmp, stricmp, strnicmp, strchr, strcspn, strspn, strprbk, atof, atoi, atol?

• strstr - пошук підрядка в рядку

• strlen - визначення довжини рядка

• strcpy - копіювання рядка

• strncpy - копіювання частини рядка

• strcat - додавання рядка до іншого

• strncat - додавання частини рядка до іншого

• strcmp - порівняння рядків

• strncmp - порівняння частин рядків

• stricmp, strnicmp - порівняння рядків без врахування регістру

• strchr - пошук символу в рядку

• strcspn - визначення довжини початкової частини рядка, яка не містить символи з іншого рядка

• strspn - визначення довжини початкової частини рядка, яка містить символи з іншого рядка

• strpbrk - пошук першого символу в рядку, який міститься в іншому рядку

• atof - перетворення рядка на дійсне число

• atoi - перетворення рядка на ціле число

• atol - перетворення рядка на довге ціле число

10. Яке призначення, синтаксис опису і оголошення структур (struct) С/С++? Чим вони відрізняються від масивів?

Структура - це складений тип даних, який об'єднує змінні різних типів під одним ім'ям.

Оголошення:

struct імя\_структури {

тип\_даних1 поле1;

тип\_даних2 поле2;

...

} [імя\_екземпляра];

11. Що таке член (поле) структури struct С/С++ та який оператор реалізовує доступ до нього?

Член структури в мові програмування С/С++ - це змінна або об'єкт, який є частиною структури. Доступ до нього здійснюється за допомогою оператора крапки (.).

12. Що у С/С++ називають дескриптором структури (struct)? За якими правилами він утворюється і задля чого використовується програмістом?

Дескриптор структури (struct) у С/С++ - це ім'я самої структури, яке програміст обирає. Він формується за правилами ідентифікаторів у мові С/С++. Дескриптор використовується для звернення до структури в коді програми.

13. Що слід виконати (записати у вихідному коді програми) для оголошення змінної типу структура, якщо означений тип описано у заголовковому файлі?

Для оголошення змінної типу структура, якщо тип описано у заголовковому файлі, необхідно включити цей заголовок за допомогою директиви #include, а потім оголосити змінну з використанням ім'я структури, за яким слідує ім'я змінної.

14. Мовою програмування С/С++ наведіть приклад оголошення структури (struct), один із членів якої ― структура.

Прикоад:

struct InnerStruct {

int x;

float y;

};

struct OuterStruct {

int a;

InnerStruct inner;

};

15. Наведіть синтаксис оголошення змінної множинного типу та перелічіть допустимі операції над нею. Чим з погляду використання відрізняються змінні типу масив і типу множина?

Оголошення:

• Масив: Оголошується з використанням квадратних дужок і вказується розмір масиву. Наприклад, int myArray[10];

• Множина: Оголошується з використанням типу set з бібліотеки STL (стандартної бібліотеки шаблонів). Наприклад, set<int> mySet;

Операції:

• Масив: Операції над масивом включають доступ за індексом, присвоєння значення за індексом, ітерацію через елементи за допомогою циклу.

• Множина: Операції над множиною включають додавання, видалення, перевірку наявності, отримання розміру та очищення.

Відмінності:

1. Упорядкованість:

Масив зберігає елементи в заданому порядку, тоді як множина зберігає їх в відсортованому порядку.

2. Дублікати:

Масив допускає дублікати елементів, тоді як множина автоматично унікальна, тобто не може містити однакові елементи.

3. Розмір:

Розмір масиву фіксований при оголошенні і не може змінюватися, тоді як множина може динамічно змінювати свій розмір.

4. Доступ до елементів:

Масив забезпечує прямий доступ до елементів за допомогою індексів, тоді як для доступу до елементів множини використовуються вищезгадані операції (insert(), erase() тощо).

16. Що у С/С++ розуміють під явним і неявним перетворення типів та яким чином реалізовується кожен з них?

Явне перетворення типів (explicit type casting) - це коли ви явно вказуєте, в який тип потрібно перетворити дане значення. Здійснюється за допомогою оператора приведення типу: (тип\_даних)вираз

Приклад:

int x = 5;

double y = (double)x; // явне перетворення int в double

Неявне перетворення типів (implicit type conversion) - це коли компілятор автоматично перетворює один тип даних в інший, відповідно до правил мови. Приклад:

int x = 5;

double y = x; // неявне перетворення int в doubleПравила неявного перетворення типів в C/C++ базуються на ієрархії типів даних і правилах арифметичних операцій. Компілятор намагається автоматично привести операнди до "найбільшого" типу, щоб уникнути втрати даних.

17. Наведіть приклад оголошення потокового об’єкта ofstream з заголовкового файла fstream С++. Яким чином за його допомогою здійснюється запис даних у файловий потік (файл)?

Приклад:

#include <fstream>

using namespace std;

int main() {

ofstream outputFile("output.txt");

// Запис даних у файловий потік

outputFile << "Hello, world!";

outputFile.close();

return 0;

}

18. У якому режимі відкриються файлові потоки, об’єкти яких оголошені типом fstream, ofstream, ifstream відповідно?

ifstream, відкриваються в режимі ios\_base::out, ios\_base::out та ios\_base::in відповідно.

19. Яке призначення наступних функцій-членів потокового об’єкта ifstream С++: open(), eof(), close()?

• open(): відкриває файл для читання.

• eof(): повертає true, якщо досягнута кінцева позиція в файлі.

• close(): закриває файловий потік.

20. Поясніть призначення кожної з перелічених констант режимів відкриття файлових потоків у С++: ios\_base::app, ios\_base::ate, ios\_base::in, ios\_base::out, ios\_base::trunc (простір імен – std).

• ios\_base::app: відкриває файл для запису в кінець.

• ios\_base::ate: відкриває файл, встановлюючи позицію в кінець файлу.

• ios\_base::in: відкриває файл для читання.

• ios\_base::out: відкриває файл для запису.

• ios\_base::trunc: обрізає вміст файлу, якщо файл вже існує.