

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних технологій та комп'ютерної інженерії

Г.М. Коротенко, Л.М. Коротенко

Методичні вказівки

до виконання лабораторних робіт
з дисципліни

«Обчислювальна техніка та програмування»

для бакалаврів спеціальності

141 «Електроенергетика, електротехніка та електромеханіка»

Мова програмування C++



Дніпро
2021

Г.М. Коротенко, Л.М. Коротенко. Методичні вказівки до виконання лабораторних робіт з дисципліни «Обчислювальна техніка та програмування» для бакалаврів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка». Мова програмування C++ / – Дніпро: НТУ «ДП», 2021. – 179 с.

Укладачі:

Г.М. Коротенко, д-р техн. наук

Л.М. Коротенко, канд. техн. наук

Методичні матеріали призначені для виконання лабораторних робіт з дисципліни «Обчислювальна техніка та програмування» бакалаврами спеціальності 141 «Електроенергетика, електротехніка та електромеханіка».

Методичні вказівки включають 12 лабораторних робіт. У них розглянуто практичні і методичні аспекти процесу навчання студентів комплексу навичок, пов'язаних з використанням мови C++, яка сьогодні вважається однією з найбільш популярних і поширених мов програмування (МП).

C++ – це МП загального призначення, створена Б'ярне Страуструпом як розширення мови програмування C, або "C з класами". C++ легко поєднує об'єктно-орієнтовані методи з загальними методами шаблонів, створюючи сучасну потужну мову програмування, корисну для вирішення проблем у незліченній кількості областей. Остання еволюція C++ з C++ 17 на C++ 20 принесла ще більші вдосконалення цій багатій мові. Оскільки C++ стає ще більш виразним, все більша кількість розробників вбудованих систем відкриває для себе нові способи використання цієї багатогранної мови та можливості для створення продуктивного та ефективного програмного забезпечення для багаточисельних видів мікроконтролерів. Окрім того дана МП підтримує програмування вбудованих систем на мікроконтролерах в режимі реального часу.

Поточним часом МП C++ використовується в наступних галузях.

- Програмне забезпечення для баз даних.
- Вбудовані системи реального часу (ВСПЧ), (Embedded System, ES).
- Телекомунікаційні системи, мережеве обладнання (комутатори, маршрутизатори, ADSL модеми тощо).
- Побутова електроніка (стільникові телефони, КПК, ігрові консолі, цифрові фотоапарати, електрочайники, мікрохвильові печі, посудомийні машини та ін.).
- Транспортна автоматика.
- Системи телемеханіки, системи моніторингу, навігації, спостереження, бортові системи для військових і космічних застосувань.
- Інтернет речей (Internet of Things, IoT). Включає наступні рівні: розумні речі, розумні послуги, розумні будинки, розумні міста.

Зміст

Введення.....	7
Лабораторна робота №1. Основи роботи з числовими типами даних в C++.....	8
1.1. Числові типи в C++.....	8
1.2. Цілочисловий тип даних <code>int</code>	8
1.3. Операція присвоювання.....	9
1.4. Імена змінних в C++.....	9
1.5. Арифметичні операції для десяткових типів чисел в C++.....	10
1.6. Завдання для роботи з цілими числами	11
1.7. Приклад розв'язання завдання до лабораторної роботи.....	13
1.8. Дійсний тип даних (з плаваючою точкою) в C++.....	14
1.9. Додаткові операції присвоювання в C++.....	15
1.10. Унарні операції інкремента і декремента	16
1.11. Математичні функції в C++.....	17
1.12. Завдання до роботи з цілими й дійсними числами.....	19
1.13. Приклад розв'язання завдання до лабораторної роботи.....	22
Лабораторна робота №2. Робота з управляючими операторами розгалуження <code>if</code> та <code>if...else</code> в C++.....	25
2.1. Булева алгебра.....	25
2.2. Управляючий оператор розгалуження <code>if</code> в мові C++.....	28
2.3. Управляючий оператор розгалуження <code>if...else</code> в мові C++.....	30
2.4. Завдання до роботи з управляючими операторами розгалуження.....	32
2.5. Приклад розв'язання завдання до лабораторної роботи.....	36
Лабораторна робота №3. Робота з управляючим оператором вибору <code>switch</code> в C++.....	39
3.1. Призначення управляючого оператора розгалуження множинного вибору <code>switch</code>	39
3.2. Загальна форма опису структури оператора вибору <code>switch</code>	39
3.3. Приклад використання управляючого оператора розгалуження множинного вибору <code>switch</code>	40
3.4. Вкладені управляючі оператори <code>switch</code>	41
3.5. Завдання до роботи з управляючим оператором розгалуження множинного вибору <code>switch</code>	41
3.6. Приклад розв'язання завдання до лабораторної роботи.....	44

Лабораторна робота №4. Основи роботи з управляючими операторами циклів <code>for</code> , <code>while</code> і <code>do...while</code> . Обчислення рядів і добутків.....	45
4.1. Поняття циклу в мові програмування.....	45
4.2. Види управляючих операторів циклу у мові C++.....	45
4.3. Управляючий оператор циклу <code>for</code> . Загальна форма оператора.....	45
4.4. Приклади використання управляючого оператора циклу <code>for</code>	46
4.5. Завдання до роботи з управляючим оператором <code>for</code>	47
4.6. Приклад розв'язання завдання до лабораторної роботи.....	49
4.7. Ще один приклад розв'язання завдання до лабораторної роботи.....	50
4.8. Управляючий оператор <code>while</code> . Загальна форма.....	51
4.9. Приклад використання управляючого оператора циклу <code>while</code>	52
4.10. Загальна форма управляючого оператора циклу <code>do...while</code>	53
4.11. Приклад використання управляючого оператора циклу <code>do...while</code> ...	53
4.12. Завдання до роботи з управляючим оператором циклу <code>while</code>	54
4.13. Приклад розв'язання завдання до лабораторної роботи.....	55
Лабораторна робота №5. Основи роботи одновимірними та двовимірними масивами в C++.....	58
5.1. Визначення одновимірних та багатовимірних масивів.....	58
5.2. Опис одновимірного масиву.....	58
5.3. Способи опису одновимірних масивів.....	58
5.4. Ввод та вивід одновимірних масивів.....	59
5.5. Завдання до роботи з одновимірними масивами.....	60
5.6. Приклад розв'язання завдання до лабораторної роботи.....	62
5.7. Опис двовимірного масиву.....	64
5.8. Способи опису двовимірних масивів.....	64
5.9. Ввод та вивід двовимірних масивів.....	65
5.10. Завдання до роботи з двовимірними масивами.....	66
5.11. Приклад розв'язання завдання до лабораторної роботи.....	68
Лабораторна робота №6. Робота зі одновимірними та двовимірними динамічними масивами в C++.....	71
6.1. Визначення динамічних одновимірних та багатовимірних масивів....	71
6.2. Робота з покажчиками і динамічними масивами.....	71
6.3. Завдання до роботи з одновимірними динамічними масивами.....	73
6.4. Приклад розв'язання завдання до лабораторної роботи.....	75
6.5. Робота з двовимірними динамічними масивами в C ++.....	77

6.6. Завдання до роботи з двовимірними динамічними масивами.....	78
6.7. Приклад розв'язання завдання до лабораторної роботи.....	79
Лабораторна робота №7. Робота з функціями, які повертають і не повертають значення в C++.....	84
7.1. Визначення функцій.....	84
7.2. Опис функцій, які повертають значення.....	84
7.3. Використання функцій, які повертають значення.....	85
7.4. Завдання до роботи з функціями, які повертають значення.....	86
7.5 Приклад розв'язання завдання до лабораторної роботи.....	92
7.6. Опис функцій, які не повертають значення.....	95
7.7. Використання функцій, які не повертають значення.....	96
7.7.1. Функції, які дійсно не повертають значення.....	96
7.7.2. Передача у функцію змінних за значенням й за посиланням.....	97
7.7.3. Передача одновимірних і двовимірних масивів у функції, які не повертають значення.....	99
7.8. Завдання до роботи з функціями, які не повертають значення.....	100
7.9. Приклад розв'язання завдання до лабораторної роботи.....	103
Лабораторна робота №8. Основи роботи з символами та рядками в C++	108
8.1. Визначення символів та рядків в C++.....	108
8.2. Робота з символами та рядками в C++.....	108
8.3. Завдання до роботи з символьними масивами.....	111
8.4. Приклад розв'язання завдання до лабораторної роботи.....	113
Лабораторна робота №9. Основи роботи з функціями для роботи з рядками в C++.....	115
9.1. Визначення функції. Переваги застосування функцій.....	115
9.2. Функції для роботи з рядками.....	115
9.2.1. Визначення довжини рядків. Функція <code>strlen(S)</code>	115
9.2.2. Копіювання рядків. Функція <code>strcpy(str1, str2)</code>	115
9.2.3. Копіювання частини рядка. Функція <code>strncpy(str1, str2, n)</code> ...	116
9.2.4. Приєднання рядка. Функція <code>strcat(str1, str2)</code>	117
9.2.5. Приєднання частини рядка. Функція <code>strncat(str1, str2, n)</code> ...	117
9.2.6. Функції перетворення символів та перевірки діапазону.....	117
9.2.7. Пошук символів.....	119
9.2.7.1. Функція <code>strchr()</code>	119
9.2.7.2. Функція <code>strspn()</code>	120
9.2.8. Пошук підрядків. Функція <code>strstr()</code>	121

9.2.9. Перетворення рядка на число.....	122
9.2.9.1. Функція <code>atoi()</code>	122
9.2.9.2. Функція <code>atof()</code>	123
9.2.10. Перетворення числа на рядок.....	124
9.2.10.1. Функція <code>itoa()</code>	124
9.2.10.2 Функція <code>gcvt()</code>	125
9.3. Завдання до роботи з функціями роботи з рядками.....	126
9.4. Приклад розв'язування завдання до лабораторної роботи.....	128
Лабораторна робота №10. Основи роботи з текстовими файлами в C++...	132
10.1. Загальні поняття про файл.....	132
10.2. Особливості роботи з текстовими файлами в C++.....	132
10.3. Форматування інформації при введенні в текстовий файл.....	134
10.4. Завдання до роботи з текстовими файлами.....	135
10.5. Приклад розв'язання завдання до лабораторної роботи.....	139
Лабораторна робота №11. Основи роботи зі структурами в C++.....	142
11.1. Визначення структур.....	142
11.2. Опис структур.....	142
11.3. Передача структур в функції.....	143
11.4. Завдання до роботи зі структурами.....	145
11.5. Приклад розв'язання завдання до лабораторної роботи.....	153
Лабораторна робота №12. Робота з бінарними файлами в C++.....	157
12.1. Загальні поняття про бінарні файли.....	157
12.2. Особливості роботи з бінарними файлами.....	157
12.3. Завдання до роботи з бінарними файлами.....	159
12.4. Приклад розв'язання завдання до лабораторної роботи.....	165
Додаток 1. Приклад титульного листа до лабораторних робіт.....	168
Додаток 2. Підготовка до лабораторних робіт на C++. Встановлення і запуск Code::Blocks.....	169
Додаток 3. Коди ASCII (0 – 127).....	177
Список рекомендованої літератури.....	178

ВВЕДЕННЯ

В процесі виконання комплексу лабораторних робіт студентам пропонується засвоїти наступні практичні навички програмування засобами мови C++:

1. Основи роботи з цілими типами даних (int).
2. Основи роботи з дійсними типами даних double.
3. Основи роботи з управляючими операторами розгалуження if та if...else.
4. Робота з управляючим оператором вибору switch.
5. Робота з управляючим оператором циклу for.
6. Робота з управляючим оператором циклу while.
7. Робота з одновимірними масивами.
8. Робота з двовимірними масивами.
9. Робота з одновимірними динамічними масивами.
10. Робота з двовимірними динамічними масивами.
11. Робота з функціями, які повертають значення.
12. Робота з функціями, які не повертають значення.
13. Основи роботи з символами та рядками.
14. Основи роботи з функціями для роботи з рядками.
15. Основи роботи з текстовими файлами.
16. Основи роботи зі структурами.
17. Робота з бінарними файлами.

Весь цикл робіт тематично вибудовуваний й поєднаний у 12-ти лабораторних роботах у дидактичних цілях для цілісного сприйняття студентами технології програмування засобами мови C++.

В основу лабораторних робіт покладено ідею освоєння студентами технічних підходів та навичок щодо читання чужого коду і структурного підходу до проектування та конструювання програм методом «зверху-вниз» з використанням принципу покрокової деталізації та базових структур мови програмування для розробки, кодування та відлагодження програм на основі функціональних стандартів світових фірм та організацій (зокрема Google та Stanford University). Це, зокрема, передбачає супроводження коду програм коментарями, в кількості, достатній для швидкого розуміння алгоритму програм.

Лабораторна робота №1

Тема: Основи роботи з числовими типами даних в C++

Мета: освоєння роботи з цілочисловими та дійсними типами даних.

Теоретична частина

1.1. Числові типи в C++

Увага! Перед виконанням лабораторних робіт необхідно встановити будь-яке інтегроване середовище розробки для мови програмування C++. Процес встановлення IDE (Integrated Development Environment) CodeBlocks наведено у Додатку 2.

Дії, що виконуються в C++, тісно пов'язані з обробленням значень різнотипних даних. В C++ підтримуються *вбудовані типи даних*, які дозволяють користувачу планувати свої дії при вирішенні конкретних задач. Тип даних визначає набір операцій які можуть вживатися до значень даного типу. Причому, *перш ніж використовувати деяку змінну, треба описати її тип!* Це важливо, оскільки тип даних визначає:

- внутрішнє представлення даних у пам'яті комп'ютера (формат збереження);
- розмір виділеної пам'яті для кожної змінної;
- множину значень, які можуть приймати величини цього типу;
- операції і функції, які можна застосовувати до значень цього типу.

До *вбудованих десяткових числових типів* (ДЧТ) C++ відносяться десяткові цілі числа і числа з плаваючою точкою (дійсні). Окрім них ще підтримуються такі *вбудовані типи*:

- char (символьний);
- Wchar_t (розширений символьний);
- bool (логічний).

Про них мова піде пізніше у наступних лабораторних роботах...

1.2. Цілочисловий тип даних

До цілочисельних типів даних в C++ відносяться наступні (табл. 1.1).

Таблиця 1.2

Цілочисельні типи даних

Тип даних	Кількість байт	Діапазон значень
short int	2	-32 768 ... 32 767

Тип даних	Кількість байт	Діапазон значень
unsigned short int	2	0 ... 65 535
int	4	-2 147 483 648 ... 2 147 483 647
unsigned in данит	4	0 ... 4 294 967 295
long int	4	-2 147 483 648 ... 2 147 483 647
unsigned long int	4	0 ... 4 294 967 295

1.3. Операція присвоювання

Операція присвоювання в мові програмування C++ позначається знаком «=». Операція присвоювання дозволяє присвоїти певні значення змінним у програмі. Ці дії виконуються над двома операндами, причому лівий операнд може тільки іменованій вираз, який модифікується, наприклад, змінну. Умовимось надалі символом «☞» позначати текст програми на C++, який виконується у вікні редактора CodeBlocks, а символом «☛» – результат роботи цієї програми у вікні виконання. Тоді базова операція присвоювання «=», яка дозволяє присвоїти значення правого операнда лівому операнду, у вікні редактора буде виглядати наступним чином:

☞	<pre>int x; // надаємо ім'я x змінній цілого типу x = 2 // присвоюємо їй ціле значення 2</pre>
---	---

1.4. Імена змінних в C++

Імена змінних, функцій, міток та інших об'єктів, що визначаються користувачем, називаються *ідентифікаторами*. Ідентифікатори можуть складатися як з одного, так і з декількох символів. Першим символом повинна бути *буква або знак підкреслення*, а за ним можуть стояти букви, числа або знак підкреслення. Нижче наведено декілька прикладів коректних і некоректних імен ідентифікаторів (табл. 1.2).

Таблиця 1.2

Приклади коректних і некоректних імен ідентифікаторів

Коректні	Некоректні
count	1count
high_balance	high.,balance

У C++ прописні та рядкові літери трактуються по-різному. Отже, count, Count і COUNT – це три різних ідентифікатора. Ідентифікатор не може збігатися

з ключовими словами C++ і не повинен мати таку саму назву, як функція, що вже міститься в бібліотеці C++.

1.5. Арифметичні операції для десяткових типів чисел в C++

Для десяткових типів чисел в C++ передбачено широкий спектр операцій (табл. 1.3).

Таблиця 1.3

Арифметичні операції в C++

Позначення операції	Назва операції	Алгебраїчний вираз	Приклад C++
+	додавання	$x + y$	$x + y$
-	віднімання	$x - y$	$x - y$
*	множення	$x \cdot y$	$x * y$
/	ділення	x / y (ділення цілком!)	x / y
%	ділення за модулем	$x \bmod y$ (залишок від ділення цілком)	$x \% y$

Перевіримо наведені у таблиці 1.2 операції.

➡	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int x = 5, // перша змінна x y = 3; // друга змінна y cout << "Значення змінних:" << endl; // виводимо заголовок cout << "x = " << x << endl; // виводимо змінну x cout << "y = " << y << endl; // виводимо змінну y cout << endl; // переводимо рядок cout << "Демонструємо арифметичні операції:" << endl; cout << "Додавання: x + y = " << x + y << endl; cout << "Віднімання: x - y = " << x - y << endl; cout << "Множення: x * y = " << x * y << endl; cout << "Ділення: x / y = " << x / y << endl; cout << "Ділення дрібу: y / x = " << y / x << endl; cout << "Узяття за модулем: x % y = " << x % y << endl; return 0; // успішне закінчення програми } </pre>
---	---

```

Значення змінних:
x = 5
y = 3

Демонструємо арифметичні операції:
Додавання: x + y = 8
Віднімання: x - y = 2
Множення: x * y = 15
Ділення: x / y = 1
Ділення дрібу: y / x = 0
Узяття за модулем: x % y = 2

```

Зверніть увагу на те, що при діленні дрібу отримуємо 0 (нуль), але програмісти іноді про це забувають...

1.6. Завдання для роботи з цілими числами

Кожен студент вибирає варіант завдання за номером у списку групи.

Всі вхідні і вихідні дані в завданнях цієї групи є цілими числами. Всі числа, для яких вказана кількість цифр (двозначне число, тризначне число і т. д.), вважаються позитивними. Роздруковувати проміжні результати.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Завдання
1.	Дано номер деякого року (ціле позитивне число). Визначити відповідний йому номер століття, враховуючи, що, наприклад, початком 20 століття був 1901 рік.
2.	Дано цілі додатні числа A, B, C. На прямокутнику розміру A x B розміщено максимально можлива кількість квадратів зі стороною C (без накладання). Знайти кількість квадратів, розміщених на прямокутнику, а також площу незайнятої частини прямокутника.
3.	Дан розмір файлу в байтах. Використовуючи операцію ділення цілком, знайти кількість повних кілобайт, які займає даний файл (1 кілобайт = 1024 байта).
4.	Дано цілі додатні числа A і B ($A > B$). На відрізку довжиною A розміщено максимально можлива кількість відрізків довжиною B (без накладання). Використовуючи операцію ділення цілком, знайти кількість відрізків B, розміщених на відрізку A.

№	Завдання
5.	Дано цілі додатні числа A і B ($A > B$). На відрізку довжиною A розміщено максимально можлива кількість відрізків довжиною B (без накладання). Використовуючи операцію взяття залишку від ділення цілком, знайти довжину незайнятої частини відрізка A .
6.	Дано двозначне число. Вивести спочатку його ліву цифру (десятки), а потім – його праву цифру (одиниці). Для знаходження десятків використовувати операцію ділення цілком, для знаходження одиниць – операцію взяття залишку від ділення.
7.	Дано тризначне число. Знайти суму і добуток його цифр.
8.°	Дано тризначне число. Вивести число, отримане при перестановці цифр вихідного числа в зворотньому напрямку.
9.	Дано тризначне число. Використовуючи одну операцію ділення цілком, вивести першу цифру даного числа (сотні).
10.	Дано тризначне число. Вивести спочатку його останню цифру (одиниці), а потім – його середню цифру (десятки).
11.	Дано чотиризначне число. Знайти суму і добуток його цифр.
12.	Дано тризначне число. Вивести число, отримане при прочитанні вихідного числа справа наліво.
13.	Дано тризначне число. У ньому закреслили першу зліва цифру і приписали її справа. Вивести отримане число.
14.	Дано тризначне число. У ньому закреслили першу праворуч цифру і приписали її зліва. Вивести отримане число.
15.	Дано тризначне число. Вивести число, отримане при перестановці цифр сотень і десятків вихідного числа (наприклад, 123 перейде в 213).
16.	Дано тризначне число. Вивести число, отримане при перестановці цифр десятків і одиниць вихідного числа (наприклад, 123 перейде в 132).
17.	Дано ціле число, більше 999. Використовуючи одну операцію ділення націло і одну операцію взяття залишку від ділення, знайти цифру, відповідну розряду сотень в записі цього числа.
18.	Дано ціле число, більше 999. Використовуючи одну операцію ділення націло і одну операцію взяття залишку від ділення, знайти цифру, відповідну розряду тисяч в записі цього числа.
19.	З початку доби минуло N секунд (N – ціле). Знайти кількість повних хвилин, що минули з початку доби.
20.	З початку доби минуло N секунд (N – ціле). Знайти кількість повних годин, що минули з початку доби.
21.	З початку доби минуло N секунд (N – ціле). Знайти кількість секунд, що пройшли з початку останньої хвилини.
22.	З початку доби минуло N секунд (N – ціле). Знайти кількість секунд, що пройшли з початку останньої години.
23.	З початку доби минуло N секунд (N – ціле). Знайти кількість повних хвилин, що минули з початку останньої години.

№	Завдання
24.	Дні тижня пронумеровані наступним чином: 0 – неділя, 1 – понеділок, 2 – вівторок, ..., 6 – субота. Дано ціле число K, що лежить в діапазоні 1..365. Визначити номер дня тижня для K-го дня року, якщо відомо, що в цьому році 1 січня було понеділком.
25.	Дні тижня пронумеровані наступним чином: 0 – неділя, 1 – понеділок, 2 – вівторок, ..., 6 – субота. Дано ціле число K, що лежить в діапазоні 1..365. Визначити номер дня тижня для K-го дня року, якщо відомо, що в цьому році 1 січня було четвергом.

1.7. Приклад розв’язання завдання до лабораторної роботи

Завдання:

	Дано тризначне число у вісімковій системі числення, яке було введено користувачем. Перевести це число з вісімкової системи числення в десяткову.
--	--

Текст програми мовою C++ з коментарями і результати її роботи виглядають наступним чином.

➔	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int n, // тризначне вісімкове число w, // змінна для роботи зі значенням n d1, // кількість одиниць d2, // кількість десятків d3, // кількість сотень res; // результат в десятковій системі cout << "Програма переводить введене" << endl; // виводимо cout << "тризначне число з вісімкової" << endl; // умову cout << "системи числення в десяткову." << endl; // завдання cout << "\nВведіть тризначне вісімкове число: "; cin >> n; // вводим вісімкове число w = n; // зберігаємо n у w d1 = w % 10; // отримуємо кількість одиниць w = w / 10; // відсікаємо останню цифру d2 = w % 10; // отримуємо кількість десятків w = w / 10; // відсікаємо останню цифру </pre>
---	---

	<pre> d3 = w % 10; // отримуємо кількість сотень // Переводимо число у десяткову систему числення: res = d3 * 8 * 8 + d2 * 8 + d1 * 8 * 1; // Виводимо результат обчислень на екран: cout << "\nВи ввели вісімкове число: " << n << endl; cout << "\nУ десятковій системі числення" << endl; cout << "воно дорівнює: " << res << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
С	<p>Програма переводить введені тризначне число з вісімкової системи числення в десяткову.</p> <p>Введіть тризначне вісімкове число: 638</p> <p>Ви ввели вісімкове число: 638</p> <p>У десятковій системі числення воно дорівнює: 472</p>

1.8. Дійсний тип даних (з плаваючою точкою) в C++

Стандарт C++ визначає три типи даних для збереження дійсних значень: `float`, `double` й `long double`. Типи даних з плаваючою точкою зберігаються в пам'яті комп'ютера інакше, ніж цілочисельні. Внутрішнє представлення дійсного числа складається з двох частин – мантиї і порядку. У IBM PC-сумісних комп'ютерах величини типу `float` займають 4 байти, з яких один двійковий розряд відводиться під знак мантиї, 8 розрядів під порядок і 23 під мантию.

Для величин типу `double`, що займають 8 байт, під порядок і мантию відводиться 11 і 52 розряди відповідно. Довжина мантиї визначає точність числа, а довжина порядку – його діапазон (табл. 1.4).

Таблиця 1.4

Дійсні типи в C++

Тип даних	Кількість байт	Діапазон абсолютних значень	Точність, кількість десяткових цифр
<code>float</code>	4	від 3.4E-38 до 3.4E+38	7
<code>double</code>	8	від 1.7E-308 до 1.7E+308	15
<code>long double</code>	8, 12 або 16	від 3.4e-4932 до 3.4e+4932	15, 18 чи 33

Рекомендується всюди використовувати тільки тип `double`. Робота з ним завжди ведеться швидше, менше ймовірність помітної втрати точності при

великій кількості обчислень. Тип `float` може стати у пригоді тільки для збереження великих масивів за умови, що для вирішення поставленого завдання буде достатньо цього типу.

Необхідно пам'ятати, що дійсні числа зберігаються в пам'яті комп'ютера з деякою обмеженою точністю в двійковій системі обчислення, в той час як загальноприйнятою у використанні є десяткова система обчислення. Тому багато чисел, які точно записуються в десятковій системі, у двійковій системі можна записати тільки у вигляді нескінченного дробу.

Наприклад, числа 0.3 і 0.7 представлені в комп'ютері нескінченними дробами, в той час як число 0.25 зберігається точно, так як представляє з себе ступінь двійки. У зв'язку з цим, *категорично не рекомендується* порівнювати між собою два дійсних числа на рівність, так як таке порівняння не є коректним!

У мові C++ літери `e` або `E` означають, що число 10 потрібно звести в ступінь, який слідує за цією буквою. Наприклад, $1.2 \cdot 10^4$ еквівалентно $1.2e4$, значення $5.9736 \cdot 10^{24}$ ще можна записати як $5.9736e24$.

Для чисел менше одиниці експонент може бути негативним. Наприклад, $5e-2$ еквівалентно $5 \cdot 10^{-2}$, що, в свою чергу, означає $5/10^2$ або 0.05. Маса електрону дорівнює $9.1093822e-31$ кг.

1.9. Додаткові операції присвоювання в C++

Окрім оператора присвоювання «`=`» в C++ існують також додаткові *арифметичні оператори присвоювання* (табл. 1.5).

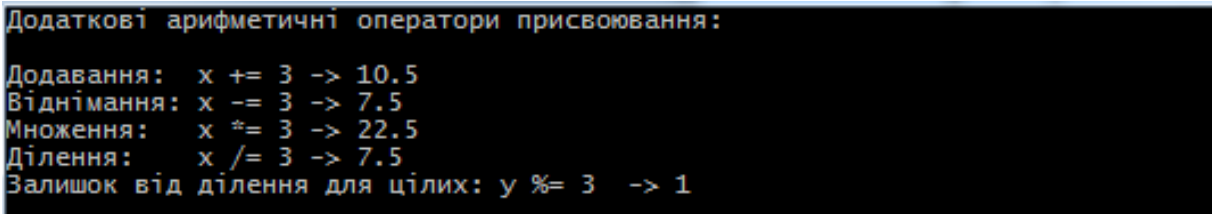
Таблиця 1.5

Додаткові арифметичні оператори присвоювання в C++

Оператор	Приклад	Аналог
<code>+=</code>	<code>x += 3</code>	<code>x = x + 3</code>
<code>-=</code>	<code>x -= 3</code>	<code>x = x - 3</code>
<code>*=</code>	<code>x *= 3</code>	<code>x = x * 3</code>
<code>/=</code>	<code>x /= 3</code>	<code>x = x / 3</code>
<code>%=</code>	<code>x %= 3</code>	<code>x = x % 3</code>
<code>**=</code>	<code>x **= 3</code>	<code>x = x ** 3</code>
<code>//=</code>	<code>x //= 3</code>	<code>x = x // 3</code>

Результат роботи додаткових арифметичних операторів присвоювання наведено нижче.

➡	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std</pre>
---	---

	<pre> int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double x = 7.5; // змінна x = 7.5 cout << "Додаткові арифметичні оператори присвоювання:" << endl; x += 3; // додаємо 3 cout << "\nДодавання: x += 3 -> " << x << endl; x -= 3; // віднімаємо 3 cout << "Віднімання: x -= 3 -> " << x << endl; x *= 3; // множимо на 3 cout << "Множення: x *= 3 -> " << x << endl; x /= 3; // ділимо на 3 cout << "Ділення: x /= 3 -> " << x << endl; int y = 5; // змінна y = 5 y %= 2; // для цілих - залишок від ділення cout << "Залишок від ділення для цілих: y %= 3 -> " << y << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
☐	 <pre> Додаткові арифметичні оператори присвоювання: Додавання: x += 3 -> 10.5 Віднімання: x -= 3 -> 7.5 Множення: x *= 3 -> 22.5 Ділення: x /= 3 -> 7.5 Залишок від ділення для цілих: y %= 3 -> 1 </pre>

Зверніть увагу на те, що залишок від ділення можна отримати тільки для цілих значень.

1.10. Унарні операції інкремента і декремента

Ці операції застосовуються *виключно з цілими змінними* і виконують наступні функції:

- *інкремент* – збільшує цілу змінну на одиницю.
- *декремент* – зменшує цілу змінну на одиницю.

Зазвичай вони можуть бути *префіксними* й *постфіксними* (табл. 1.6).

Таблиця 1.6

Операції інкремента і декремента в C++

Оператор	Вид	Приклад	Операція
Префіксний інкремент	++	++x	Інкремент x, потім обчислення x
Префіксний декремент	--	--x	Декремент x, потім обчислення x

Оператор	Вид	Приклад	Операція
Постфіксний інкремент	++	x++	Обчислення x, потім інкремент x
Постфіксний декремент	--	x--	Обчислення x, потім декремент x

Приклад дії унарних операцій інкремента і декремента наведено нижче.

➡	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int c = 5; // описуємо цілу змінну c = 5 c out << "Початкове значення c = " << c << endl; cout << "\nПрефіксний інкремент ++c = " << ++c << endl; cout << "\nПрефіксний декремент --c = " << --c << endl; cout << "\nПостфіксний інкремент c++ = " << c++ << endl; cout << "\nПісля постфіксного інкременту c = " << c << endl; cout << "\nПостфіксний декремент c-- = " << c-- << endl; cout << "\nПісля постфіксного декременту c = " << c << endl; return 0; // успішне закінчення програми } // кінець функції main</pre>
⬅	<pre>Початкове значення c = 5 Префіксний інкремент ++c = 6 Префіксний декремент --c = 5 Постфіксний інкремент c++ = 5 Після постфіксного інкременту c = 6 Постфіксний декремент c-- = 6 Після постфіксного декременту c = 5</pre>

1.11. Математичні функції в C++

Математичні функції для роботи з ДЧТ в C++ знаходяться в заголовному файлі `<cmath>`. Він пропонує збірку функцій, що дозволяють виконувати елементарні математичні обчислення.

Взагалі, всі функції в заголовному файлі `<cmath>` є глобальними для функції `main` і, таким чином, будь-яка з них викликається просто шляхом зазначення імені функції, за яким слідує укладені в круглі дужки аргументи.

Аргументами функції можуть бути константи, змінні або більш складні вирази цілого або дійсного типу. Якщо $c = 13.0$, $d = 3.0$ і $f = 4.0$, то оператор

```
cout << sqrt ( c + d * f ) << endl ;
```

обчислить і надрукує квадратний корінь з $25.0 = 13.0 + 3.0 * 4.0$, а саме 5.0 .

Список математичних функцій досить великий – його можна знайти в документації МП C++. Ми ж розглянемо нійбільш використовувані для роботи з ДЧТ (у алфавітному порядку) (табл.1.7).

Таблиця 1.7

Вбудовані математичні функції C++ для роботи з ДЧТ

Функція	Опис дії	Приклад
acos(x)	повертає значення арккосинуса	acos(1.0) = 0.0
asin(x)	повертає значення арксинуса	asin(0.0) = 0.0
atan(x)	повертає значення арктангенсу	atan(0.0) = 0.0
ceil(x)	округлення x до найменшого цілого, не меншого x	ceil(9.2) = 10.0 ceil(9.8) = 10.0
cos(x)	тригонометричний косинус x (x – у радіанах)	cos(0.0) = 1.0
exp(x)	експоненційна функція e^x	exp(1.0) = 2.718282 exp(2.0) = 7.389056
fabs(x)	абсолютне значення x	fabs(5.1) = 5.1 fabs(0.0) = 0.0 fabs(-8.76) = 8.76
floor(x)	округлення x до найбільшого цілого, яке не перевищує x	floor(9.2) = 9.0 floor(-9.8) = -10.0
fmod(x, y)	залишок від x/y, як число з плаваючою крапкою	fmod(2.6, 1.2) = 0.2
log(x)	натуральний логарифм x (за основою e)	log(2.718282) = 1.0 log(7.389056) = 2.0
log10(x)	десятковий логарифм x (за основою 10)	log10(10.0) = 1.0 log10(100.0) = 2.0
pow(x, y)	x у ступеню y	pow(2, 7) = 128.0 pow(9, 0.5) = 3.0
sin(x)	тригонометричний синус x (x у радіанах)	sin(0.0) = 0.0
sqrt(x)	корінь квадратний з x (де x – невід'ємне значення)	sqrt(9.0) = 3.0
tan(x)	тригонометричний тангенс x (x – у радіанах)	tan (0.0) = 0.0

1.12. Завдання до роботи з цілими й дійсними числами

Знайти значення функції відповідно до варіанту.

№	Завдання
1	$S = \frac{\sin^3 3x + \operatorname{arctg} \varphi - 6 \cdot 10^{3,1}}{\sqrt{ax^2 + bx + c}} + e^x \bullet \operatorname{tg}(x+2) ,$ $x = 0,21; \varphi = 4,3; a = 2,15; b = 3; c = 4,8$
2	$Z = \frac{\ln^2 x^3 + 0,375d}{e^{x^2} \sqrt{c+a}} + \frac{\operatorname{tg}^2(\pi/6) - \sqrt[5]{x^3}}{ 0,316 \cdot b \cdot c - a^2 },$ $a = 12,6; c = 1,21; d = 1,26; x = 1,13; \pi = 3,14$
3	$L = \frac{6,2 \cdot 10^{2,7} \cdot \operatorname{tg}(\pi - x^3)}{e + \ln^{1/2} \arccos b^3 } + \operatorname{arctg} \frac{10^2 \sqrt{a}}{2x + 87,2},$ $x = 3,81; b = 2,7 \cdot 10^{-2}; a = 50; \pi = 3,14$
4	$V = \cos(24 \frac{\pi}{2}) + \frac{\operatorname{tg}^5 \ln x^3 + 4,2 \cdot 10^{-2,8}}{\sqrt{ x + e^j }};$ $\varphi = 2,14; \pi = 3,142; x = 10; j = 0,5$
5	$R = e^{ \sin \pi^2 } \bullet \frac{\ln(1 + \cos^2 a) + \operatorname{tg} \sin x + 4,8 \cdot 10^{1,3}}{5\sqrt{\varphi + \gamma}};$ $x = -4,3; a = 13; \pi = 3,14; \gamma = 14 \cdot 10^{-4}; \varphi = 8$
6	$Z = \operatorname{tg} \frac{e^t + 3\varphi}{\sqrt{ t^2 + 2 }} - \frac{\cos^3 \varphi + 2,8 \cdot 10^{-3,4} + x}{\sqrt[5]{\sin^3(\pi - 4) + 1,2}};$ $t = 2,14; x = 10; a = 4,8 \cdot 10^{-2}; \pi = 3,142; \varphi = 0,51$
7	$K = x^4 \sqrt[4]{\varphi - a} + \frac{\ln^3 \sin b + \sqrt[3]{\operatorname{tg} x}}{2,3 \lg x };$ $x = 8,12; \varphi = 10; a = 7,5 \cdot 10^{-3}; b = 3,8 \cdot 10^{1,8}$

№	Завдання
8	$F = \arccos\left(\frac{\sin x/2}{\sqrt[3]{b-1}}\right) + \frac{\ln(e^\gamma + 4,1 \cdot 10^{2,1})}{\operatorname{tg}(\pi/4 + x/2)};$ $x = 2,11; \pi = 3,142; b = 19; \gamma = 0,64$
9	$Y = \frac{4,1 \cdot 10^{-1,7}}{(x - \pi) \cdot \sin 5x} + \frac{\operatorname{tg}^3 x - \lg \sqrt{a + \varphi}}{e^\pi};$ $x = -4,87; \pi = 3,14; \varphi = -2,9 \cdot 10^{-3}; a = 10$
10	$\gamma = e^{\operatorname{tg}^2 x} + \frac{\sqrt{ t - z }}{\cos^3 \pi^2 + e^\pi z^2} + 2,3 \cdot 10^{1,6};$ $\pi = 3,142; t = 2 \cdot 10^3; x = 14; z = -1,24$
11	$F = \operatorname{ctg} \left \frac{\ln \varphi + 16,3}{\sqrt{e^x - a^{3/2}}} \right + \frac{6,8 + \cos(\pi - t^2)}{\sin^3 a};$ $x = 8,6; a = 20; \varphi = 6,9; t = 0,5; \pi = 3,142$
12	$L = \frac{6,2 \cdot 10^{2,7} + \operatorname{tg}(\pi - x^3)}{e^{x/a} + \ln b^2 } + \operatorname{arctg} \frac{10^3 \cdot \sqrt{a}}{2x - b};$ $x = -3,81; \pi = 3,14; a = 220; b = 2,74 \cdot 10^2$
13	$S = \frac{e^{\arcsin} + \sqrt{ \gamma + x }}{\cos^3 \ln x^3 + 3,7 \cdot 10^{-4,2}} + \operatorname{tg}^3 (\pi/2 + \varphi);$ $\pi = 3,142; x = 20; a = 3,2 \cdot 10^{-3}; \gamma = 0,7; \varphi = 4,28$
14	$m = \frac{\cos^2 \pi - 2\varphi }{\operatorname{tg}^3 \ln t^3 + 3,2 \cdot 10^{-2,8}} + \ln \left \sqrt{\frac{e^t + x}{a}} \right ,$ $t = 3,68; \quad x = 15; \quad a = 8,2 \cdot 10^{-2}; \quad \varphi = -0,63;$ $\pi = 3,142$

№	Завдання
15	$L = \frac{\ln z \cdot \sin^{1/2} \varphi}{\alpha + \sqrt{\operatorname{tg}^2 x}} + 1,3 \cdot 10^{-1,6} \cdot \sqrt{z+x} + e^\pi;$ $x = 12,8; \quad \varphi = 14; \quad \alpha = 3,6 \cdot 10^{-4}; \quad z = -0,13; \quad \pi = 3,14$
16	$P = \sin \frac{ x }{2} \cdot \frac{\sqrt[5]{\gamma+1}}{\arccos t} + b^{\sin \gamma} \cdot \ln \operatorname{tg}(\pi/4 + 2x) ,$ $x = 6,75; \quad t = 11 \cdot 10^{-4,7}; \quad \gamma = 0,73; \quad b = 48; \quad \pi = 3,14$
17	$\gamma = \ln = \frac{\cos \sqrt{\pi \cdot x + a^2}}{x^2 + e^\pi} + \frac{4,7 \cdot 10^{-4,2}}{\cos^2 \varphi + \sqrt[3]{\operatorname{tg} x^2}},$ $x = -2,93; \quad \pi = 3,14; \quad a = 10; \quad \varphi = -31,4 \cdot 10^{-3}$
18	$\gamma = \frac{\arccos(\sqrt[5]{ z }) + 6,9 \cdot 10^{3,3}}{t + \sqrt[5]{\ln t+1 }} + e \cdot \operatorname{arctg} \sqrt{x+\pi},$ $\pi = 3,142; \quad t = 2 \cdot 10^3; \quad x = 24; \quad z = -3,72$
19	$\nu = \lg \sin(x - \pi/2) + \frac{\arcsin(\varphi - 8,7)^3 + 3,16 \cdot 10^{2,7}}{\sqrt{\alpha^2 + r} + \operatorname{tg}^2(5x + \varphi)},$ $x = 2,367; \quad \varphi = 9,2; \quad \alpha = 21; \quad \pi = 3,14159; \quad r = 47 \cdot 10^3$
20	$p = \frac{x^5}{e^2} + \sqrt{\frac{\cos^3 a}{1 + \varphi}} + \frac{6,8 \cdot 10^{1,9} \cdot \operatorname{ctg}(\pi - \varphi)}{\ln \operatorname{tg}(x - \pi/2) } + r,$ $x = 8,441; \quad a = 14; \quad r = 3,2 \cdot 10^3; \quad \varphi = 2,7; \quad \pi = 3,141$
21	$\gamma = \arcsin \frac{e^x - \cos a}{10^3 + 3^t} + \ln \left \frac{\sqrt{\sin^{1/5} + \pi}}{2,3 \operatorname{tg}^3(a-b)} \right ,$ $x = 4,5; \quad a = 10; \quad b = 2,7 \cdot 10^{-2,3}; \quad t = 0,5; \quad \pi = 3,142$

№	Завдання
22	$\alpha = \ln 1+a + \frac{e^{\pi-x}}{3,2 \cdot 10^{1,6} \cdot \cos \ln b } + \arccos^{1/2} \left x \cdot 10^{-2} \right ,$ $x = -4,18; \quad \pi = 3,14; \quad a = 150; \quad b = 3,2 \cdot 10^3$
23	$V = \operatorname{tg}^5 \left \ln x^3 \right + \frac{2,4 \cdot 10^{-3,1} + \arccos(a + \pi/4)}{\sqrt{e^{\sin \gamma} + \lg \varphi + x }},$ $\varphi = 2,28; \quad \pi = 3,142; \quad x = 11; \quad a = 2,7 \cdot 10^{-3}; \quad \varphi = 0,5$
24	$R = \frac{\lg \left \cos \sqrt{a^2 + b} \right }{e^{\sin x}} + \frac{1 + \arcsin b + 3,6 \cdot 10^{2,7}}{\operatorname{tg} \lg^{1/2} x \cdot \pi },$ $x = -3,8; \quad a = 13; \quad b = 2,7 \cdot 10^{-3}; \quad \pi = 3,14$
25	$Z = e^{\sqrt{ t^2 + 0,1 }} + \frac{2,4 \cdot 10^{-4,2} + \operatorname{tg}^3 \ln(x-4)}{\sin^{1/3}(\pi + 4)} + \frac{3^x}{\arccos 4 },$ $t = 2,36; \quad x = 10; \quad a = 4,2 \cdot 10^{-3}; \quad \varphi = -0,28; \quad \pi = 3,142$

1.13. Приклад розв'язання завдання до лабораторної роботи

Завдання: обчислити задану формулу:

$F = \frac{e^x \cdot \sqrt{ t+1 }}{\sin b} + \frac{\arcsin \gamma + b \cdot \sqrt{x} + 4,2 \cdot 10^{2,2}}{\lg \cos(\pi/4 + x/2) },$ $x = 2,72; \quad t = -12 \cdot 10^{-2,8}; \quad \gamma = 0,64; \quad b = 19; \quad \pi = 3,141$

У свій час, відомий комп'ютерний фахівець Брайан Керніган (співавтор Денніса Рітчі по книзі «Мова програмування С») сказав, що управління складністю є сутністю комп'ютерного програмування на шляху запобігання зайвих помилок у програмі. Тому, ми поділяємо складний вираз F на три частини: $f1$, $f2$ та $f3$. Програма стає простіше, її легше читати, відлагоджувати та супроводжувати.

Текст програми з результатами її роботи.



```
#include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
#include <cmath>        // підключаємо математичну бібліотеку
using namespace std;   // оголошуємо простір імен std

int main() // функція main виконується першою
{
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // Описуємо змінні:
    double x,          // аргумент функції

           t,          // змінні, які
           gamma,      // приймають участь
           b,          // у формулі

           f1,         // допоміжні
           f2,         // змінні
           f3,         // до розрахунку

           F;          // значення функції

    cout << "Обчислюємо задану формулу.\n" << endl;
    cout << "Ініціюємо необхідні змінні:" << endl;
    x = 2.72;
    cout << "x      = " << x << endl;
    t = -12 * pow(10, -2.8);
    cout << "t      = " << t << endl;
    gamma = 0.64;
    cout << "gamma = " << gamma << endl;
    b = 19;
    cout << "b      = " << b << endl;

    // Обчислюємо складові основної формули:
    f1 = (exp(x) * sqrt(abs(t - 1))) / sin(b);
    f2 = asin(gamma) + b * sqrt(x) + 4.2 * pow(10, 2.2);
    f3 = log10(abs(cos(3.141 + 0.5 * x)));

    // Обчислюємо основну формулу:
    F = f1 + f2 / f3;

    // Виводимо на екран результат:
    cout << "\nРезультат роботи програми:" << endl;
    cout << "F = " << F << endl;

    return 0; // успішне закінчення програми
}             // кінець функції main
```

```
Обчислюємо задану формулу.  
Ініціюємо необхідні змінні:  
x      = 2.72  
t      = -0.0190187  
gamma  = 0.64  
b      = 19  
  
Результат роботи програми:  
F = -926.552
```

Увага! Коли Ви пишете свою програму, тримайтеся правила: *не робити довгих рядків*. Тут потрібно дотримуватися такого стилю, щоб при читанні тексту програми не треба було повертати голову зліва направо й назад, а слідити тільки очами, як при читанні поеми...

Пам'ятайте, що автор мови C++ Б'ярне Страуструп сказав: «Найбільш фундаментальна проблема в розробці програмного забезпечення – складність. Є тільки один основний спосіб впоратися зі складністю: розділяй і володарюй».

Лабораторна робота №2

Тема: Робота з управляючими операторами розгалуження if та if...else в C++

Мета: освоєння практичних навичок роботи з управляючими структурами прийняття рішень і булевою алгеброю.

Теоретична частина

2.1. Булева алгебра

У більшості програм на певному кроці їх виконання потрібно вибирати той чи інший варіант подальших дій, тобто приймати рішення. Для цього в C++ використовуються *управляючі структури* if та if...else, тобто логічні структури які керують порядком виконання операторів в програмі. Оператор if здійснює перевірку булевого виразу, щоб визначити, чи є він істинним або хибним. Вирази, які перевіряються оператором if, називаються *булевими виразами* на честь англійського математика Джорджа Буля. У 1800-х роках Буль винайшов математичну систему, в якій абстрактні поняття істинності і хибності могли використовуватися в обчисленнях. Для вибору того чи іншого варіанту рішень в обчисленнях є виконання деякого *булевого виразу* з подальшою оцінкою його істинності чи хибності. Тому умова у операторі if повинна бути булевого типу, тобто її результат повинен мати лише два значення:

0 (false – «хибність») або 1 (true – «істина»)

Як правило, *логічний вираз* (булевий вираз), що перевіряється оператором if, формується *оператором порівняння* (реляційним оператором). Оператор порівняння визначає, чи існує між двома значеннями певне відношення. Оператори порівняння МП C++ представлені у таблиці 2.1.

Таблиця 2.1

Оператори порівняння в C++

Позначення операторів порівняння	Математичне позначення операторів	Назва опертора	Приклад
==	=	дорівнює	x == y
!=	≠	не дорівнює	x != y

Позначення операторів порівняння	Математичне позначення операторів	Назва оператора	Приклад
>	>	більше ніж	$x > y$
<	<	менше ніж	$x < y$
>=	\geq	більше або дорівнює	$x \geq y$
<=	\leq	менше або дорівнює	$x \leq y$

Приклад виконання операторів порівняння та результати їх виконання виглядають наступним чином.

➔	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int x = 5, // задаємо x y = 3; // задаємо y cout << "Оператори порівняння та" << endl; cout << "приклади їх виконання:\n" << endl; cout << "5 == 3 -> " << (x == y) << endl; cout << "5 != 3 -> " << (x != y) << endl; cout << "5 > 3 -> " << (x > y) << endl; cout << "5 < 3 -> " << (x < y) << endl; cout << "5 >= 3 -> " << (x >= y) << endl; cout << "5 <= 3 -> " << (x <= y) << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
⌂	<pre> Оператори порівняння та приклади їх виконання: 5 == 3 -> 0 5 != 3 -> 1 5 > 3 -> 1 5 < 3 -> 0 5 >= 3 -> 1 5 <= 3 -> 0 </pre>

Але, якщо необхідно сконструювати складний логічний вираз, то прості логічні вирази треба об'єднувати у складні з допомогою *логічних операторів* (таблиця 2.2).

Логічні оператори в C++

Позначення логічних операторів	Математичне позначення операторів	Назва операції	Приклад
&&	\wedge	логічне множення, І, AND	$x > 3 \ \&\& \ x < 10$
	\vee	логічне додавання, АБО, OR	$x > 3 \ \ x < 4$
!	\neg	заперечення, NOT	$!(x > 3 \ \&\& \ x < 10)$

Приклад виконання логічних операторів та результати їх виконання виглядають наступним чином.

➔	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int x = 5; // задаємо x cout << "Логічні оператори та" << endl; cout << "приклади їх виконання:\n" << endl; cout << "x > 3 && x < 10 -> " << (x > 3 && x < 10) << endl; cout << "x > 3 or x < 4 -> " << (x > 3 x < 4) << endl; cout << "!(x > 3 && x < 10) -> " << !(x > 3 && x < 10) << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
⬅	<pre> Логічні оператори та приклади їх виконання: x > 3 && x < 10 -> 1 x > 3 or x < 4 -> 1 !(x > 3 && x < 10) -> 0 </pre>

При обчисленні результатів дії логічних операторів в булевій алгебрі прийняті наступні угоди:

– логічне множення (кон'юнкція) – це функція двох змінних $f_1(x, y) = x \wedge y$ (читається: « x і y »), яка виражає висловлювання, котре істинне тільки в тому випадку, якщо істинні обидва висловлювання x і y (табл.2.3);

– логічне додавання (диз'юнкція) – це функція двох змінних $f_2(x, y) = x \vee y$ (читається: « x або y »), яка виражає висловлювання, котре істинне тоді і тільки тоді, коли, принаймні, одне з висловлювань x або y є істинним (табл.2.3);

– заперечення – це функція одного змінного $f_3(x) = \neg x$ (читається: «не x »), яка виражає висловлювання, котре істинне, якщо висловлювання x хибно, и хибно, якщо x істинне (табл.2.3).

Таблиця 2.3

Результати дії логічних операторів в булевій алгебрі

Логічне множення – and			Логічне додавання – or			Заперечення – not	
x	y	$x \wedge y$	x	y	$x \vee y$	x	$\neg x$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Таким чином, щоб створити логічний вираз визначення попадання змінної x у діапазон $[0; 4]$ треба створити логічний оператор $x \geq 0 \ \&\& \ x \leq 4$, а якщо змінна x у цей діапазон не попадає, то створимо такий оператор: $x < 0 \ || \ x > 4$.

Оператор заперечення, як правило, використовується в складних логічних виразах для спрощення логічних виразів.

2.2. Управляючий оператор розгалуження if в мові C++

Дотепер ми використовували тільки найпростіший тип структур: неуправляючу структуру – *послідовну структуру*, тобто структуру з послідовним виконанням операторів. Тобто *послідовна структура* в програмі є набором операторів, які виконуються в тому порядку, в якому вони з'являються.

Але, все ж таки, головними в програмі є *управляючі оператори (структури)*. *Управляючий оператор* – це логічний елемент програми для обробки даних, який управляє порядком виконання (або невиконання) набору зазначених операторів, званих *блоком* (рис 2.1).

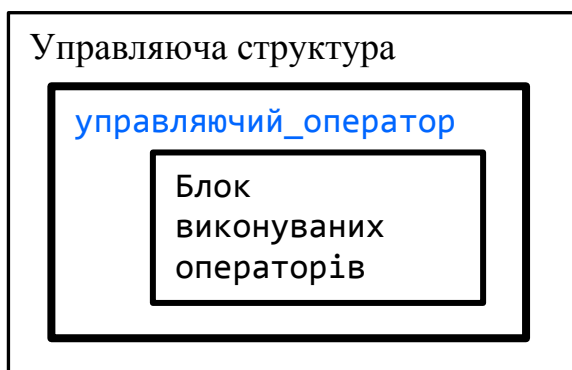


Рис. 2.1. Склад управляючої структури

До складу управляючої структури входять: *управляючий оператор* та *блок виконуваних операторів*. Цей блок може мати в собі від одного до потрібного числа операторів мови C++ у тому числі – й *управляючих*! Ознакою блока є хоча б один відступ вправо всіх операторів блока. Але згідно документу «Керівництво зі стилю Google C++», який описує угоду компанії Гугл про те, як писати код для мови C++, ми будемо дотримуватися відступу з *чотирьох пробілів*! ЗАБОРОНЯЄТЬСЯ використання клавіші Tab!

При такому підході – *він називається структурним* – Ви у своїй уяві *проектуйте* і *конструюєте* програму, а на екрані комп'ютера *реалізуєте* її, використовуючи слідування операторів і комбінації управляючих структур.

Отже, управляючий оператор (управляюча структура) розгалуження `if` у мові програмування C++ має такий загальний вигляд.

```
// Попередній оператор
if ( умова )
{
    Блок операторів,
    які виконуються,
    якщо умова = True,
    у тому числі
    управляючих.
}
// Подальший оператор
```

Зверніть увагу, що блок операторів розміщений у *фігурних дужках*, перед кожним з них зроблено *відступ* у чотири пробіли! Якщо блок складається з *одного оператора*, то дужки можна опустити. Умова у операторі `if` становить собою *логічний вираз* будь-якої складності. Блок виконується, якщо `умова = True`. Нижче наведено приклад виконання управляючого оператора.


```
➡ #include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
using namespace std;    // оголошуємо простір імен std

int main()    // функція main виконується першою
{
    SetConsoleOutputCP(1251);    // локалізуємо вивід

    // Описуємо змінні:
    int a = 10,    // ініціалізуємо змінну для умови
        c = 60,    // ініціалізуємо змінну для присвоювання
        b = 100;    // ініціалізуємо змінну для присвоювання

    cout << "Перевіряємо роботу управляючого" << endl;
    cout << "оператора розгалуження if:\n" << endl;

    if ( a == 10 )    // перевіряємо умову
```

	<pre> b = 300; // присвоюємо значення змінній cout << "a == 10 -> " << (a == 10) << endl; // значення умови cout << "b = " << b << endl; // значення змінної if (a > 10) // перевіряємо умову { // відкриваюча дужка блока b = 600; // присвоюємо значення змінній b c = 700; // присвоюємо значення змінній c } // закриваюча дужка блока cout << "\na > 10 -> " << (a > 10) << endl; // значення умови cout << "b = " << b << endl; // значення змінної b cout << "c = " << c << endl; // значення змінної c return 0; // успішне закінчення програми } // кінець функції main </pre>
	<pre> Перевіряємо роботу управляючого оператора розгалуження if: a == 10 -> 1 b = 300 a > 10 -> 0 b = 300 c = 60 </pre>

2.3. Управляючий оператор розгалуження if...else в мові C++

Управляючий оператор if...else комбінується з двох елементів: управляючого оператора розгалуження if та частки – else.

<pre> // Попередній оператор if (умова) { Блок операторів, які виконуються, якщо умова = True, у тому числі управляючих. } else { Блок операторів, які виконуються, якщо умова = False, у тому числі управляючих. } // Подальший оператор </pre>
--

Умова у операторі if...else становить собою *логічний вираз* будь-якої складності. Блок виконується, якщо умова=True. Якщо блок складається з *одного оператора*, то дужки можна опустити.

Приклад перевірки дії управляючої операторія if...else.

➔	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int a = 10, // ініціалізуємо змінну для умови b = 100; // ініціалізуємо змінну для перевірки cout << "Перевіряємо роботу управляючого" << endl; cout << "оператора розгалуження if...else:\n" << endl; if (a == 10) // перевіряємо умову b = 300; // присвоюємо значення змінній else // гілка else b = 600; // присвоюємо значення змінній cout << "a == 10 -> " << (a == 10) << endl; // значення умови cout << "b = " << b << endl; // значення змінної a = 5; // ініціалізуємо змінну для умови if (a > 10) // перевіряємо умову b = 300; // присвоюємо значення змінній else // гілка else b = 600; // присвоюємо значення змінній cout << "\na == 5 -> " << (a == 5) << endl; // значення умови cout << "b = " << b << endl; // значення змінної return 0; // успішне закінчення програми } // кінець функції main </pre>
⌂	<pre> Перевіряємо роботу управляючого оператора розгалуження if/else: a == 10 -> 1 b = 300 a == 5 -> 1 b = 600 </pre>

2.4. Завдання до роботи з управляючими операторами розгалуження

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти значення функції відповідно до варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Завдання
1.	$y = \begin{cases} \sin x, & \text{при } x \leq 0; \\ \arctg x, & \text{при } 0 < x \leq \pi/4; \\ \log_2 x, & \text{при } \pi/4 < x \leq 32; \\ 1/x, & \text{в інших випадках.} \end{cases}$
2.	$z = \begin{cases} \ln x , & \text{при } x < -\pi; \\ \sin x + \cos 2x, & \text{при } -\pi \leq x < \pi; \\ x^3 + 1, & \text{при } \pi \leq x < 10; \\ \frac{x+1}{x^2+8}, & \text{при } 10 \leq x < 100; \\ \ln x, & \text{в інших випадках.} \end{cases}$
3.	$z = \begin{cases} \arctg \frac{x}{y}, & \text{при } y \neq 0, x > y ; \\ \arcsin \frac{x}{y}, & \text{при } y \neq 0, x \leq y ; \\ 0, & \text{в інших випадках.} \end{cases}$
4.	$z = \begin{cases} \frac{1}{x} + \frac{1}{y}, & \text{при } x < -10, y < -5; \\ \frac{x-y}{x+y}, & \text{при } -10 \leq x < 0, \\ & -5 \leq y < 0; \\ \frac{\sin x}{\cos y}, & \bullet \text{ при } 0 \leq x < 2\pi, \\ & 0 \leq y < \pi/2; \\ \ln(x^2 + y^2), & \text{в інших випадках.} \end{cases}$

№	Завдання
5.	$z = \begin{cases} \sin \frac{x+y}{y}, & \text{при } y \neq 0, x > y ; \\ \arccos \frac{x}{y}, & \text{при } y \neq 0, x \leq y ; \\ \pi, & \text{в інших випадках.} \end{cases}$
6.	$y = \begin{cases} 0, & \text{при } x \leq 0; \\ 1/x, & \text{при } 0 < x \leq 1; \\ x^2, & \text{при } 1 < x \leq 4; \\ 14 + \log_2 x, & \text{при } x > 4. \end{cases}$
7.	$z = \begin{cases} \frac{2}{x} - \frac{4}{y}, & \text{при } x < -20, y < -10; \\ \frac{x-y-2}{x+y}, & \text{при } -20 \leq x < 0, \\ & -10 \leq y < 0; \\ \frac{\sin x + \cos x}{\cos y}, & \text{при } 0 \leq x < 2\pi, \\ & 0 \leq y < \pi/2; \\ \log_4(x^2 + y^2), & \text{в інших випадках.} \end{cases}$
8.	$y = \begin{cases} \cos x, & \text{при } x \leq 0; \\ \arcsin x, & \text{при } 0 < x \leq \pi/2; \\ \log_4 x, & \text{при } \pi/2 < x \leq 64; \\ 1/x^2, & \text{в інших випадках.} \end{cases}$
9.	$y = \begin{cases} 0, & \text{при } x < 0; \\ 1/(x^2 + 1), & \text{при } 0 \leq x \leq 1; \\ x^3, & \text{при } 1 < x \leq 4; \\ 62 + \log_8 x, & \text{при } x > 4. \end{cases}$
10.	$z = \begin{cases} x + y, & \text{при } x < 0, y \leq 0; \\ \operatorname{arctg} \frac{x}{y}, & \text{при } 0 \leq x < 10, \\ & y > 0; \\ 0, & \text{в інших випадках.} \end{cases}$
11.	$z = \begin{cases} \sin x + \sin y, & \text{при } x < 0, y > 0; \\ \cos x + \cos y, & \text{при } x > 0, y < 0; \\ \frac{\sin x + \cos y}{\cos^2 x + \sin^2 y}, & \text{в інших випадках.} \end{cases}$

№	Завдання
12.	$y = \frac{\text{sign } k \cdot \text{sign } l \cdot \delta_{kl}}{\text{sign}(k+l)\text{sign}(k-l)\delta_{k+l,k-l}},$ <p>де</p> $\text{sign } x = \begin{cases} -1, & \text{при } x < 0; \\ 0, & \text{при } x = 0; \\ 1, & \text{при } x > 0; \end{cases}$ $\delta_{nm} = \begin{cases} 1, & \text{при } n = m; \\ 0, & \text{в інших випадках.} \end{cases}$
13.	$z = \begin{cases} \ln x, & \text{при } x < -\pi; \\ \sin x - \cos x, & \text{при } -\pi \leq x < \pi; \\ x + 1, & \text{при } \pi \leq x < 10; \\ \frac{x+3}{x^2+8}, & \text{при } 10 \leq x < 100; \\ \log_2 x, & \text{в інших випадках.} \end{cases}$
14.	$z = \begin{cases} \text{ctg } \frac{x}{y}, & \text{при } y \neq 0, x > y ; \\ \arccos n \frac{x}{y}, & \text{при } y \neq 0, x \leq y ; \\ -1, & \text{в інших випадках.} \end{cases}$
15.	$y = \begin{cases} \cos x, & \text{при } x \leq 0; \\ \text{tg } x, & \text{при } 0 < x \leq \frac{\pi}{4}; \\ \log 10x, & \text{при } \frac{\pi}{4} < x \leq 32; \\ \frac{1}{x^2}, & \text{в інших випадках.} \end{cases}$
16.	$y = \begin{cases} x^2 + a & a > b, b > 2 \\ \max(a, b) & a < b, b < 2 \\ \sin^3 x^2 + \sqrt[3]{a^b - \text{tg } b} & a = b \\ \min(a, b) & \text{в інших випадках.} \end{cases}$
17.	$z = \begin{cases} \text{ctg } \frac{x}{y}, & \text{при } y \neq 0, x > y ; \\ \arccos n \frac{x}{y}, & \text{при } y \neq 0, x \leq y ; \\ -1, & \text{в інших випадках.} \end{cases}$

№	Завдання
18.	$y = \begin{cases} -100, & \text{при } x \leq 0; \\ \sqrt[3]{x^2}, & \text{при } 0 < x \leq 1; \\ x, & \text{при } 1 < x \leq 4; \\ 100 + \ln x, & \text{при } x > 4. \end{cases}$
19.	$z = \begin{cases} \frac{1}{x^2} + \frac{1}{y^2}, & \text{при } x < -5, y < -1; \\ \frac{x-y}{x+y}, & \text{при } -5 \leq x < 0, \\ & -1 \leq y < 0; \\ \frac{\cos x}{\sin y}, & \text{при } 0 \leq x < 2\pi, \\ & 0 \leq y < \pi/2; \\ \ln(x+y), & \text{в інших випадках.} \end{cases}$
20.	$z = \begin{cases} \sin \frac{x+y}{y^2}, & \text{при } y \neq 0, x > y; \\ \arcsin \frac{x}{y}, & \text{при } y \neq 0, x \leq y; \\ \pi/2, & \text{в інших випадках.} \end{cases}$
21.	$k = \begin{cases} \frac{2}{x} - \frac{4}{y}, & \text{при } x < -10, y < -5; \\ \frac{x-y-2}{x+y}, & \text{при } -10 \leq x < 0, \\ & -5 \leq y < 0; \\ \frac{\sin x + \cos x}{\sin y}, & \text{при } 0 \leq x < 2\pi, \\ & 0 \leq y < \pi/2; \\ \ln(x^2 + y^2), & \text{в інших випадках.} \end{cases}$
22.	$w = \begin{cases} \pi, & \text{при } x < 0; \\ \sqrt[3]{x^3}, & \text{при } 0 \leq x \leq 3; \\ x^2, & \text{при } 3 < x \leq 10; \\ 6,2 + \log_5 x, & \text{при } x > 10. \end{cases}$
23.	$y = \begin{cases} \sin x, & \text{при } x \leq 0; \\ \arccos x, & \text{при } 0 < x \leq \pi/2; \\ \log_4 x, & \text{при } \pi/2 < x \leq 64; \\ \sqrt[3]{x^2}, & \text{в інших випадках.} \end{cases}$

№	Завдання
24.	$v = \begin{cases} x * y, & \text{при } x < 0, y \leq 0; \\ \operatorname{ctg} \frac{x}{y}, & \text{при } 0 \leq x < 10, 0 < y \leq 9; \\ -100, & \text{в інших випадках.} \end{cases}$
25.	$z = \begin{cases} \arcsin x + \sin y, & \text{при } x < 0, \\ & y > 0; \\ \cos x^2 - \cos y^2, & \text{при } x > 0, \\ & y < 0; \\ \frac{\sin x + \cos y}{\cos^2 x + \sin^2 y}, & \text{в інших випадках.} \end{cases}$

2.5. Приклад розв'язання завдання до лабораторної роботи

	$\phi = \begin{cases} x^3 - y, & \text{при } x < 0, y \leq 0; \\ \cos \frac{x}{y}, & \text{при } 0 \leq x < 8, y \leq 12; \\ -88, & \text{в інших випадках.} \end{cases}$
--	---

Як бачимо, тут потрібно використовувати управляючий оператор **if...else**. Але у завданні у нас не дві гілки, а три. Тому, у гілку **else** вставляємо ще один оператор **if...else** й отримуємо управляючий оператор розгалуження виду.

```
// Попередній оператор
if ( умова )
{
    Блок операторів,
    які виконуються,
    якщо умова = True,
    у тому числі
    управляючих.
}
else
    if ( умова )
    {
        Блок операторів,
        які виконуються,
        якщо умова = True,
        у тому числі
        управляючих.
    }
    else
    {
        Блок операторів,
        які виконуються,
        якщо умова = False,
        у тому числі
        управляючих.
    }
// Подальший оператор
```

Зверніть увагу, що відступи у операторі розгалуження підкреслюють *рівні вкладеності блоків* у керуючому операторі розгалуження!

За необхідністю, вставок **if...else** може бути стільки, скільки необхідно. Звісно, кожен програміст знає, що ті управляючі оператори, які він використовує, повинні працювати так, як він собі уявляє. Але, якщо це не так – буде велика біда... Тому, ми повинні уважно перевірити роботу цього оператора.

Текст програми з результатами її роботи наведено нижче.

➔	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cmath> // підключаємо математичну бібліотеку using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double x, // змінна для x перевірки y, // змінна для y перевірки fi; // змінна для результату // Виводимо завдання: cout << "Знайти значення функції fi відповідно" << endl; cout << "до варіанту, використовуючи управляючі" << endl; cout << "оператори розгалуження:\n" << endl; // Введення значень з клавіатури: cout << "Введіть значення x = "; // запрошуємо до вводу x cin >> x; // вводимо x cout << "Введіть значення y = "; // запрошуємо до вводу y cin >> y; // вводимо y // Конструкція управляючої структури розгалуження: if (x < 0 && y <= 0) // перевіряємо 1-шу умову { fi = pow(x, 3) - y; // обчислюємо fi cout << "Гілка 1\n"; // друкуємо, якщо ми тут } else if (0 <= x && x < 8 && y <= 12) // перевіряємо 2-гу умову { fi = cos(x / y); // обчислюємо fi cout << "Гілка 2\n"; // друкуємо, якщо ми тут } else // інакше! { fi = 88; // обчислюємо fi cout << "Гілка 3\n"; // друкуємо, якщо ми тут } }</pre>
---	--

	<pre> } cout << "Результат: fi = " << fi << endl; // виводимо результат return 0; // успішне закінчення програми } // кінець функції main </pre>
☞	<p>Знайти значення функції f_i відповідно до варіанту, використовуючи управляючі оператори розгалуження:</p> <p>Введіть значення $x = -1.5$ Введіть значення $y = -0.5$ Гілка 1 Результат: $f_i = -2.875$</p>

Результат для другої гілки буде таким.

☞	<p>Знайти значення функції f_i відповідно до варіанту, використовуючи управляючі оператори розгалуження:</p> <p>Введіть значення $x = 5.7$ Введіть значення $y = 9.15$ Гілка 2 Результат: $f_i = 0.81216$</p>
---	--

Результат для третьої гілки.

☞	<p>Знайти значення функції f_i відповідно до варіанту, використовуючи управляючі оператори розгалуження:</p> <p>Введіть значення $x = 9.14$ Введіть значення $y = 11.64$ Гілка 3 Результат: $f_i = 88$</p>
---	---

Пам'ятайте. Автор мови C++ Б'ярне Страуструп сказав: «C++ розроблений, щоб дозволити вам висловлювати ідеї, але якщо у вас немає ідей або ви не знаєте, як їх висловити, C++ не пропонує великої допомоги».

Лабораторна робота №3

Тема: Робота з управляючим оператором вибору switch в C++

Мета: освоєння практичних навичок роботи з управляючим оператором розгалуження множинного вибору switch.

Теоретична частина

3.1. Призначення управляючого оператора розгалуження множинного вибору switch

Управляючий оператор розгалуження множинного вибору switch може бути *замінений набором операторів if\else*. Однак, у деяких випадках використання оператора switch може бути більш ефективним.

3.2. Загальна форма опису структури оператора вибору switch

Спочатку йде ключове слово switch й в дужках вираз, з яким буде виконуватися порівняння у кожній з його гілок. Зазвичай цей вираз являє собою тільки одну змінну, наприклад z, або константу – «2», або символ «с», але це може бути і щось більш складне, наприклад, $nX+2$ або $nX-nY$. Єдине обмеження до цього виразу – воно має бути цілого або символьного типу даних (тобто типів char, short, int, long, або long long). Змінні типу з плаваючою точкою float та double використовувати не можна! Далі слідує необхідна кількість *варіантів* з мітками case і необов'язковий *варіант* з міткою default, який виконується, якщо вираз не має жодного збігу з *попередніми* мітками. Після них йде *послідовність_операторів*, яка виконується у кожному випадку.

Оператор break застосовується для *переривання виконання* оператора switch в будь-якому місці: управління при його виконанні передається на оператор, наступний відразу за оператором switch!

Оператор switch порівнює вираз з кожною з міток case. Якщо з'являється відповідність до вмісту case, виконуються оператори, наступні за цією міткою.

```
// Попередній оператор
switch ( вираз )
{
    case константа1:
        послідовність_операторів1;
        break;
    case константа2:
        послідовність_операторів2;
        break;
```

```

...

case константаN:
    послідовність_операторівN;
break;
default:
    послідовність_операторів;
}
// Подальший оператор

```

3.3. Приклад використання управляючого оператора розгалуження множинного вибору switch

Ознайомимося з роботою цієї управляючої структури на прикладі виводу назви десяткових цифр від 0 до 9.

```

➔ #include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
using namespace std;    // оголошуємо простір імен std

int main()    // функція main виконується першою
{
    SetConsoleOutputCP(1251);    // локалізуємо вивід


    // Описуємо змінну:
    int digit;    // змінна для десяткової цифри

    cout << " Програма роздруковує назви" << endl;    // виводимо
    cout << "десяткових цифр від 0 до 9:\n" << endl;    // заголовок

    cout << "Введіть десяткову цифру digit = ";    // запрошуємо до
    вводу
    cin >> digit;    // вводимо цифру

    // Виводимо назви десяткових цифр:
    switch ( digit )    // переключаємо за цифрою
    {
        case 0:    // цифра 0
            cout << " нуль" << endl;
            break;    // виходимо зі switch
        case 1:    // цифра 1
            cout << " один" << endl;
            break;    // виходимо зі switch
        case 2:    // цифра 2
            cout << " два" << endl;
            break;    // виходимо зі switch
        case 3:    // цифра 3
            cout << " три" << endl;
            break;    // виходимо зі switch
        case 4:    // цифра 4
            cout << " чотири" << endl;

```


	<pre> break; // виходимо зі switch case 5: // цифра 5 cout << " п'ять" << endl; break; // виходимо зі switch case 6: // цифра 6 cout << " шість" << endl; break; // виходимо зі switch case 7: // цифра 7 cout << " сім" << endl; break; // виходимо зі switch case 8: // цифра 8 cout << " вісім" << endl; break; // виходимо зі switch case 9: // цифра 9 cout << " дев'ять" << endl; break; default: // якщо не виконалися попередні варіанти cout << "Введений символ – не цифра!" << endl; } // кінець switch return 0; // успішне закінчення програми } // кінець функції main </pre>
	<pre> Програма роздруковує назви десятичних цифр ввід 0 до 9: Введіть десятичну цифру digit = 9 дев'ять </pre>

3.4. Вкладені управляючі оператори switch

Оператор switch може бути вкладеним в інший управляючий оператор, яким може бути:

- оператор розгалуження if;
- оператор вибору switch;
- оператор циклу for;
- оператор циклу while;
- оператор циклу do/while.

3.5. Завдання до роботи з управляючим оператором розгалуження множинного вибору switch

№	Завдання
1.	Дано ціле число в діапазоні 1..7. Вивести рядок – назву дня тижня, що відповідає даному числу (1 – «понеділок», 2 – «вівторок» і т.ін.).
2.	Дано ціле число K. Вивести рядок – опис оцінки, що відповідає числу K (1 – «погано», 2 – «незадовільно», 3 – «задовільно», 4 – «добре», 5 – «відмінно»). Якщо K не лежить в діапазоні 1..5, то

№	Завдання
	вивести рядок «помилка».
3.	Дано номер місяця – ціле число в діапазоні 1..12 (1 – січень, 2 – лютий і т.ін.). Вивести назву відповідної пори року («зима», «весна», «літо», «осінь»).
4.	Дано номер місяця – ціле число в діапазоні 1..12 (1 – січень, 2 – лютий і т. ін.). Визначити кількість днів у цьому місяці для високосного року.
5.	Арифметичні дії над числами пронумеровані в такий спосіб: 1 – складання, 2 – віднімання, 3 – множення, 4 – поділ. Дано номер дії N (ціле число в діапазоні 1..4) і дійсні числа A і B (B не дорівнює 0). Виконати над числами вказану дію і вивести результат.
6.	Одиниці довжини пронумеровані в такий спосіб: 1 – дециметр, 2 – кілометр, 3 – метр, 4 – міліметр, 5 – сантиметр. Дано номер одиниці довжини (ціле число в діапазоні 1..5) і довжина відрізка в цих одиницях (дійсне число). Знайти довжину відрізка в метрах.
7.	Одиниці маси пронумеровані в такий спосіб: 1 – кілограм, 2 – міліграм, 3 – грам, 4 – тонна, 5 – центнер. Дано номер одиниці маси (ціле число в діапазоні 1..5) і маса тіла в цих одиницях (дійсне число). Знайти масу тіла в кілограмах.
8.	Скласти програму, яка реалізувала б такі дії: по введеному числу K (до 10) видавала б відповідну їй римську цифру.
9.	Скласти програму, яка б по введеному числу (до 10) видавала б назву відповідної їй римської цифри.
10.	Робот може переміщатися в чотирьох напрямках («10» – північ, «20» – захід, «30» – південь, «40» – схід) і приймати три цифрові команди: «0» – продовжувати рух, «1» – поворот наліво, «-1» – поворот на право. Дано число C – вихідний напрямок роботи і число N – послана йому команда. Вивести напрямок роботи після виконання отриманої команди.
11.	Локаатор орієнтований на одну зі сторін світла («10» – північ, «20» – захід, «30» – південь, «40» – схід) і може приймати три цифрові команди повороту: «1» – поворот наліво, «-1» – поворот направо, «2» – поворот на 180°. Дано число C – вихідна орієнтація локатора і числа N1 і N2 – дві послані команди. Вивести орієнтацію локатора після виконання цих команд.
12.	Елементи окружності пронумеровані в такий спосіб: 1 – радіус R, 2 – діаметр $D = 2 \cdot R$, 3 – довжина $L = 2 \cdot \pi \cdot R$, 4 – площа кола $S = \pi \cdot R^2$. Дано номер одного з цих елементів і його значення. Вивести значення інших елементів даної окружності (в тому ж порядку).
13.	Елементи рівнобедреного прямокутного трикутника пронумеровані в такий спосіб: 1 – катет a, 2 – гіпотенуза $c = 2 \cdot a$, 3 – висота h, опущена на гіпотенузу $h = c / 2$, 4 – площа $S = c \cdot h / 2$. Дано номер одного з цих елементів і його значення. Вивести значення інших

№	Завдання
	елементів даного трикутника (в тому ж порядку).
14.	Елементи рівностороннього трикутника пронумеровані в такий спосіб: 1 – сторона a , 2 – радіус $R1$ вписаного кола $R1 = \frac{a}{2\sqrt{3}}$, 3 – радіус $R2$ описаного кола $R2 = \frac{a}{\sqrt{3}}$, 4 – площа $S = \frac{\sqrt{3}}{4}a^2$. Дано номер одного з цих елементів і його значення. Вивести значення інших елементів даного трикутника (в тому ж порядку).
15.	Мастям гральних карт присвоєно порядкові номери: 1 – піки, 2 – трефи, 3 – бубни, 4 – черви. Вартісті карт, старших десяти, привласнені номери: 11 – валет, 12 – дама, 13 – король, 14 – туз. Дано два цілих числа: N – вартість ($6 \leq N \leq 14$) і M – масть карти ($1 \leq M \leq 4$). Вивести назву відповідної карти виду «шістка бубон», «дама червей», «туз треф» і т.ін.
16.	Дано ціле число в діапазоні 20..69, визначальне вік (в роках). Вивести рядок – опис зазначеного віку, забезпечивши правильне узгодження числа з словом «рік», наприклад: 20 – «двадцять років», 32 – «тридцять два роки», 41 – «сорок один рік».
17.	Дано ціле число в діапазоні 10..40, яке визначає кількість навчальних завдань по деякій темі. Вивести рядок – опис зазначеної кількості завдань, забезпечивши правильне узгодження числа зі словами «навчальне завдання», наприклад: 18 – «вісімнадцять навчальних завдань», 23 – «двадцять три навчальних завдання», 31 – «тридцять одне навчальне завдання».
18.	Дано ціле число в діапазоні 100..999. Вивести рядок – опис даного числа, наприклад: 256 – "двісті п'ятдесят шість", 814 – "вісімсот чотирнадцять».
19.	У східному календарі прийнятий 60-річний цикл, що складається з 12-річних підциклів, що позначаються назвами кольорів: зелений, червоний, жовтий, білий і чорний. У кожному підциклі роки носять назви тварин: щура, корови, тигра, зайця, дракона, змії, коня, вівці, мавпи, курки, собаки і свині. За номером року визначити його назву, наприклад, 1984 рік – початок циклу: «рік зеленого щура».
20.	Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату. Вивести знак Зодіаку, що відповідає цій даті: «Водолій» (20.1–18.2), «Риба» (19.2–20.3), «Овен» (21.3–19.4), «Телець» (20.4–20.5), «Близнюки» (21.5– 21.6), «Рак» (22.6–22.7), «Лев» (23.7–22.8), «Діва» (23.8–22.9), «Терези» (23.9–22.10), «Скорпіон» (23.10–22.11), «Стрілець» (23.11–21.12), «Козеріг» (22.12–19.1).

№	Завдання
21.	Одиниці довжини пронумеровані в такий спосіб: 1 – дециметр, 2 – кілометр, 3 – метр, 4 – міліметр, 5 – сантиметр. Дано номер одиниці довжини (ціле число в діапазоні 1..5) і довжина відрізка в цих одиницях (дійсне число). Знайти довжину відрізка в метрах.
22.	Одиниці маси пронумеровані в такий спосіб: 1 – кілограм, 2 – міліграм, 3 – грам, 4 – тонна, 5 – центнер. Дано номер одиниці маси (ціле число в діапазоні 1..5) і маса тіла в цих одиницях (дійсне число). Знайти масу тіла в кілограмах.
23.	Скласти програму, яка б за порядковим номером місяця видавала б його назву.
24.	Скласти програму, яка б за порядковим номером місяця видавала до якого часу року він належить.
25.	Для цілого числа K від 1 до 9 надрукувати фразу "мені K років", враховуючи при цьому, що при деяких значеннях K слово "років" треба замінити на слово "рік" або "роки".

3.6. Приклад розв'язання завдання до лабораторної роботи

Вищенаведені завдання треба виконати згідно прикладу з пункту 2.2. Перевірити три мітки case й мітку default.

Пам'ятайте. Автор мови C++ Б'ярне Страуструп сказав: «Найважливішим аспектом розробки програмного забезпечення є чітке розуміння того, що ви намагаєтеся створити».

Лабораторна робота №4

Тема: Основи роботи з управляючими операторами циклів `for`, `while` і `do...while`. Обчислення рядів і добутків

Мета: освоєння роботи з циклами `for`, `while` і `do...while` у мові C++. Використання циклів для обчислення рядів і добутків.

Теоретична частина

4.1. Поняття циклу в мові програмування

Часто, при програмуванні задач, потрібно, щоб одна й та ж послідовність команд виконувалась декілька разів. Для цього у мовах програмування застосовується поняття циклічного процесу або циклу. Алгоритм, в якому певна послідовність команд повторюється декілька разів з новими вхідними даними називається циклічним.

Циклічний процес організовується за допомогою управляючих операторів циклу. Мова програмування C/C++ має в наявності зручні для роботи оператори організації циклічних процесів обробки даних.

4.2. Види управляючих операторів циклу у мові C++

У мові C++ існує три види управляючих операторів циклу:

- цикл `for`;
- цикл `while` з передумовою;
- цикл `do...while` з постумовою.

Кожен з операторів циклу має свої особливості застосування. Будь-який з вищенаведених операторів циклу може бути замінений іншим.

4.3. Управляючий оператор циклу `for`. Загальна форма оператора

Управляючий оператор циклу `for`, також званий циклом з параметром, забезпечує багаторазове виконання оператора або блока операторів, у тому числі й управляючих. Цикл управляється виразом, який ітерується, і має наступний синтаксис.

```
// Попередній оператор
for ( ініціалізація; вираз; приріст )
{
    Блок операторів,
    які виконуються,
    в тому числі -
    управляючих
}
// Подальший оператор
```

де ініціалізація – операція присвоювання початкового значення управляючій змінній циклу. Ця змінна є лічильником, який керує роботою відповідного циклу.

вираз – умовний вираз, в якому перевіряється значення змінної циклу;

приріст – вираз, за яким обчислюється значення змінної циклу для кожної ітерації.

Цикл `for` виконується до тих пір, поки значення вираз дорівнює `true`. Як тільки значення вираз дорівнює `false`, виконання циклу припиняється і виконується оператор, що слідує за циклом `for`.

Тут потрібно додати, що окрім ініціалізації в операторі циклу ще здійснюється *опис типу управляючої змінної* – `int`. Змінну можна описувати й в голові програми, але вже стало стандартом описувати її в управляючому операторі `for`.

4.4. Приклади використання управляючого оператора циклу `for`

Наступні приклади показують способи зміни управляючої змінної циклу в операторі `for`.

а) Зміна управляючої змінної циклу від 1 до 10.

➔	<pre>// Організовуємо цикл for: for (int i = 1; i <= 10; i++) cout << " " << i ; // виводимо значення змінної циклу</pre>
⌂	<pre>1 2 3 4 5 6 7 8 9 10</pre>

б) Зміна управляючої змінної циклу від 10 до 1 з кроком -1 (зі зменшенням на 1).

➔	<pre>// Організовуємо цикл for: for (int i = 10; i >= 1; i--) cout << " " << i ; // виводимо значення змінної циклу</pre>
⌂	<pre>10 9 8 7 6 5 4 3 2 1</pre>

в) Зміна управляючої змінної циклу від 7 до 77 з кроком 7.

➔	<pre>// Організовуємо цикл for: for (int i = 7; i <= 70; i += 7) cout << " " << i ; // виводимо значення змінної циклу</pre>
⌂	<pre>7 14 21 28 35 42 49 56 63 70</pre>

г) Зміна управляючої змінної циклу від 20 до 2 з кроком -2.

➡	<pre>// Організовуємо цикл for: for (int i = 20; i >= 2 ; i -= 2) cout << " " << i ; // виводимо значення змінної циклу</pre>
⬅	<pre>20 18 16 14 12 10 8 6 4 2</pre>

д) Зміна управляючої змінної циклу у наступній послідовності: 2, 5, 8, 11, 14.

➡	<pre>// Організовуємо цикл for: for (int j = 2; j <= 20; j += 3) cout << " " << j ; // виводимо значення змінної циклу</pre>
⬅	<pre>2 5 8 11 14 17 20</pre>

е) Зміна управляючої змінної циклу у наступній послідовності: 99, 88, 77, 66, 55, 44, 33, 22, 11, 0.

➡	<pre>// Організовуємо цикл for: for (int j = 99; j >= 0; j -= 11) cout << " " << j ; // виводимо значення змінної циклу</pre>
⬅	<pre>99 88 77 66 55 44 33 22 11 0</pre>

4.5. Завдання до роботи з управляючим оператором for

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти значення функції відповідно до варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

Примітка : Σ – сума , Π – добуток.

№	Завдання
1.	$S = \sum_{i=2}^7 \left(\frac{(i-1)!}{\cos ix} + \ln x \right)$
2.	$S = \sum_{i=3}^{10} \frac{i! + \cos x}{xi! + \operatorname{tg} ix}$

№	Завдання
3.	$P = \prod_{i=1}^5 \frac{i! + \sqrt{i!+1}}{ x-i! }$
4.	$S = \sum_{i=1}^5 \frac{y^2 (i+1)! + x}{(x+1)(i+1)!}$
5.	$P = \prod_{i=2}^5 (\cos^2 ix - \sin x^2)(i-1)!$
6.	$S = \sum_{i=1}^7 (\sqrt[i]{i!+y} - z^i)$
7.	$S = \sum_{i=5}^{15} \frac{i! - 0.576}{i! + \ln i!}$
8.	$P = \prod_{t=3}^6 (-1)^{t-1} \frac{(t-1)! + \sqrt[5]{t^2+x}}{e^t + t^2}$
9.	$S = \sum_{k=2}^{15} (-1)^k \frac{k^2}{k!-1} \cos(xk!)$
10.	$P = \prod_{t=1}^4 e^{t!/5} \frac{2}{e^t + t!}$
11.	$z = \sum_{j=1}^5 \frac{A}{A^{(j+1)!} + B} ((j+1)! - AB)$
12.	$P = \prod_{k=1}^6 \frac{e^k}{(k+1)!} \arctan^2 k$
13.	$P = \prod_{t=2}^5 (-1)^{t-1} \frac{t! + x^2}{tg^2 x^3 - xt!}$
14.	$P = \prod_{i=2}^9 \frac{\sqrt[3]{i} - \sqrt{i!}}{tg x^2 - y}$
15.	$S = \sum_{n=1}^5 (-1)^{n+1} \frac{2n!-1}{2n!} x^{2n!}$
16.	$z = \sum_{i=1}^5 (-1)^{i!} \frac{(x+a)b}{b^2 - (x+a)^x}$
17.	$S = \sum_{k=2}^5 k^k \frac{\sqrt[k]{x+k!}}{\ln k + k!}$
18.	$P = \prod_{i=2}^9 (-1)^i \frac{(i+1)! + \ln i}{i^2 - (i+1)!}$

№	Завдання
19.	$S = \sum_{i=1}^5 x^{-i} \frac{i! + \cos(ix)}{x^3 + \cos x}$
20.	$S = \sum_{i=2}^4 (i-1)! \frac{e^i + (i-1)!}{i^2}$
21.	$S = \sum_{t=1}^4 \left(\ln x^{t!} - x^{t!} - \sqrt[t]{x^2} \right)^2$
22.	$S = \sum_{i=3}^7 x^{-i} \frac{(i-1)!}{i^2 + \sin^2 x}$
23.	$P = \prod_{t=3}^6 (-1)^t \frac{t! + \lg t-x }{y^2 + \ln y-t }$
24.	$S = 10 \sum_{i=1}^5 x^i (\cos x^2 + i! \sin^2 x)$
25.	$P = \prod_{t=1}^5 \sqrt[t]{a} \frac{(t+1)! \ln a}{(t+1)! + e^a}$

4.6. Приклад розв'язання завдання до лабораторної роботи

	$S = \sum_{i=2}^4 (i-1)! \frac{e^i + (i-1)!}{i^2}$
--	--

Зверніть увагу, що вираз $(i-1)!$ Використовується у формулі два рази. Тому ми будемо обчислювати його окремо, щоб не робити цього двічі.

➔	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cmath> // підключаємо математичну бібліотеку using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double S; // для суми int fact; // для факторіалу // Виводимо завдання: cout << "Знайти суму ряду відповідно до варіанту" << endl; cout << "використовуючи управляючий оператор for: \n" << endl; S = 0.0; // обнуляємо комірку для суми </pre>
---	---

	<pre> fact = 1; // готуємо початкове значення факторіалу (2-1)! = 1! // Організовуємо цикл for: for (int i = 2; i <= 4; i++) // цикл від 2 до 4 { S += fact * (exp(i) + fact) // обчислюємо суму / (i * i); // замість ступеня обчислюємо добуток fact *= i ; // готуємо факторіал } cout << "Результат: S = " << S << endl; // виводимо результат return 0; // успішне закінчення програми } // кінець функції main </pre>
☞	<p>Знайти суму ряду відповідно до варіанту використовуючи управляючий оператор for:</p> <p>Результат: S = 29.7295</p>


Взагалі, намагайтеся підмічати елементи ряду, які повторюються і обчислюйте їх заздалегідь.

4.7. Ще один приклад розв'язання завдання до лабораторної роботи

	$S = \sum_{n=1}^5 \frac{2n-1}{n^2(n+1)^2}$
--	--

Якщо уважно подивитися на цей ряд, можна побачити, що у чисельнику маємо послідовність непарних чисел, а у знаменнику у дужках – послідовність натуральних чисел, починаючи з двійки. Це треба урахувати й не програмувати дріб у лоб! Тому програма буде мати такий вигляд.

☞	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cmath> // підключаємо математичну бібліотеку using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double S, // для суми numerat, // для чисельника denom; // для знаменника // Виводимо завдання: </pre>
---	--

	<pre> cout << "Знайти суму ряду відповідно до варіанту," << endl; cout << "використовуючи управляючий оператор for: \n" << endl; S = 0.0; // обнуляємо комірку для суми numerat = 0; // готуємо змінну для чисельника denom = 1; // готуємо змінну для знаменника // Організовуємо цикл for: for (int i = 1; i <= 5; i++) // цикл від 1 до 5 { numerat += 1; // обчислюємо значення для чисельника denom += 1; // обчислюємо значення для знаменника S += numerat / (i * i * denom * denom); // обчислюємо суму } cout << "Результат: S = " << S << endl; // виводимо результат return 0; // успішне закінчення програми } // кінець функції main </pre>
	<pre> Знайти суму ряду відповідно до варіанту використовуючи управляючий оператор for: Результат: S = 0.341944 </pre>

В алгоритмі програми ми отримали деяку економію операцій на обчисленні змінної S в циклі тому, що відмовились від двох функцій піднесення у ступінь Pow у тому, що степенювання обчислюються через ряди, а множення працює скоріше.

Також у даному прикладі, щоб отримати дійсне значення S , замість цілих типів (`int`) для змінних `numerat` і `denom` ми вибрали дійсний тип `double`.

Пам'ятайте. Автор мови C++ Б'ярне Страуструп сказав: «Якщо ви думаєте, що це просто, значить, ви неправильно зрозуміли проблему».

4.8. Управляючий оператор `while`. Загальна форма

Цикл `while` називається циклом з передумовою. Загальна форма циклу `while` наступна.

```

// Попередній оператор
while ( умова )
{
    Блок операторів,
    які виконуються,
    в тому числі -
    управляючих
}
// Подальший оператор

```

де умова – становить собою *логічний вираз* будь-якої складності, в якому перевіряється значення змінної циклу. На цьому етапі визначається подальше виконання циклу.

Блок операторів виконується до тих пір, поки умова повертає значення `true`. Як тільки вираз стає рівним `false`, виконання циклу `while` припиняється і управління передається наступному за циклом `while` оператору. Увага! Якщо перед входом в цикл `while` – умова повертає значення `false`, то не буде виконано жодної дії!

Як правило, цикл while використовується тоді, коли неможливо визначити точне значення кількості проходів при виконанні циклу.

4.9. Приклад використання управляючого оператора циклу while

Розробимо програму, яка виводить п'ять цілих чисел.

➡	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cmath> // підключаємо математичну бібліотеку using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінну: int counter = 0; // для суми // Виводимо завдання: cout << "Вивести п'ять цілих чисел: \n" << endl; while (counter < 5) // перевірка умови продовження циклу { counter++; // інкрементуємо змінну cout << "counter = " << counter << endl; // виводимо змінну } return 0; // успішне закінчення програми } // кінець функції main</pre>
Ⓒ	<pre>Вивести п'ять цілих чисел: counter = 1 counter = 2 counter = 3 counter = 4 counter = 5</pre>

4.10. Загальна форма управляючого оператора циклу do...while

Цикл do...while доцільно використовувати у випадках, коли ітерацію потрібно зробити *хоча б один* раз. На відміну від циклів for та while, у циклі do...while умова перевіряється *при виході* з циклу (а не *при вході* в цикл).

Загальна форма управляючого оператора циклу do...while.

```
// Попередній оператор
do
{
    Блок операторів,
    які виконуються,
    в тому числі -
    управляючих
}
while (умова);
// Подальший оператор
```

де умова – становить собою *логічний вираз* будь-якої складності, в якому перевіряється значення змінної циклу. На цьому етапі визначається подальше виконання циклу. Фігурні дужки у цьому циклі необов'язкові.

Цикл працює наступним чином. Спочатку відбувається виконання тіла циклу. Потім перевіряється значення конструкції умова (*логічний вираз*). Якщо значення умова є істинним (true), тіло циклу виконується знову. Як тільки значення умова прийме значення false, виконання циклу припиняється й управління передається на наступний за циклом while оператор.

4.11. Приклад використання управляючого оператора циклу do...while

Розробимо програму, яка виводить п'ять цілих чисел.

```
➔ #include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
#include <cmath>        // підключаємо математичну бібліотеку
using namespace std;   // оголошуємо простір імен std

int main() // функція main виконується першою
{
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // Описуємо змінну:
    int counter = 0;    // для суми

    // Виводимо завдання:
    cout << "Вивести п'ять цілих чисел: \n" << endl;

    do // початок циклу do...while
    {
        counter++; // інкрементуємо змінну
```

	<pre> cout << "counter = " << counter << endl; // виводимо змінну } while (counter < 5); // перевірка умови продовження циклу return 0; // успішне закінчення програми } // кінець функції main </pre>
С	<p>Вивести п'ять цілих чисел:</p> <pre> counter = 1 counter = 2 counter = 3 counter = 4 counter = 5 </pre>

4.12. Завдання до роботи з управляючим оператором циклу while

№	Варіант	Точність ε
1	$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!} + \dots$	0.001
2	$1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots \pm \frac{1}{n!} \mp \dots$	0.005
3	$1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n} + \dots$	0.0001
4	$1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots \pm \frac{1}{2^n} \mp \dots$	0.05
5	$\frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \dots + \frac{1}{(2n-1)(2n+1)} + \dots$	0.00005
6	$\frac{1}{3 \cdot 5} + \frac{1}{7 \cdot 9} + \dots + \frac{1}{(4n-1)(4n+1)} + \dots$	0.0001
7	$\frac{1}{1^2} + \frac{1}{3^2} + \dots + \frac{1}{(2n+1)^2} + \dots$	0.001
8	$1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2} + \dots$	0.005
9	$1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots \pm \frac{1}{n^4} \mp \dots$	0.0005
10	$\frac{1}{1^4} + \frac{1}{3^4} + \dots + \frac{1}{(2n+1)^4} + \dots$	0.01
11	$\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \dots + \frac{1}{(3n-2)(3n+1)} + \dots$	0.05
12	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \pm \dots$	0.005
13	$x - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} \pm \dots$	0.0001

№	Варіант	Точність ϵ
14	$\frac{x-1}{x} - \frac{(x-1)^2}{2x^2} + \dots + (-1)^n \frac{(x-1)^n}{nx^n} + \dots$	0.00001
15	$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n} \pm \dots$	0.00001
16	$x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$	0.05
17	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!} + \dots$	0.001
18	$x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} + \dots$	0.01
19	$1 + \frac{1}{2} \left(2 \cdot 3x + 3 \cdot 4x^2 + 4 \cdot 5x^3 + \dots \right)$	0.05
20	$1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \dots + \frac{(x \ln a)^n}{n!} + \dots$	0.05
21	$\frac{\pi}{2} - \left(x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{2n+1} \pm \right)$	0.00005
22	$\frac{1}{2} + \frac{3}{2^2} + \dots + \frac{2n-1}{2^n} + \dots$	0.0005
23	$1 + \frac{1}{4}x - \frac{1 \cdot 3}{4 \cdot 8}x^2 + \frac{1 \cdot 3 \cdot 6}{4 \cdot 8 \cdot 12}x^3 - \frac{1 \cdot 3 \cdot 6 \cdot 10}{4 \cdot 8 \cdot 12 \cdot 16}x^4 \pm \dots$	0.0001
24	$1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \dots \pm \frac{1}{n^2} \mp \dots$	0.001
25	$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$	0.005

4.13 Приклад розв'язання завдання до лабораторної роботи

	$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \pm \dots$	0.0001
--	---	--------

Перша думка – просто запрограмувати формулу для обчислення поточного члена ряду... Але у нас є факторіал, який може давати переповнення при великих значеннях n .

Спробуємо покроково розглянути процес зміни членів знакоперемінного ряду:

Номер шагу	1	2	3	...
Знак	+	-	+	...
Член ряду	$\frac{x}{1}$	$\frac{x * x^2}{1 * 2 * 3}$	$\frac{x^3 * x^2}{3! * 4 * 5}$...

Трошки поміркувавши, ми можемо, починаючи з другого члена ряду чисельник домножати на x^2 , а знаменник доділювати послідовно на $2*3$, $4*5$ і так далі... Зрозуміли? Тоді – до роботи.

```

➔ #include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
#include <cmath>        // підключаємо математичну бібліотеку
using namespace std;    // оголошуємо простір імен std

int main()    // функція main виконується першою
{
    SetConsoleOutputCP(1251);    // локалізуємо вивід

    // Описуємо змінні:
    int n = 0;    // кількість обертів циклу
    double S,    // сума ряду
           x,    // аргумент
           eps,    // точність обчислень
           member,    // поточний член ряду
           divider;    // дільник для членів ряду

    // Виводимо завдання:
    cout << "Програма обчислює суму елементів " << endl;
    cout << "числового ряду з заданою точністю eps. \n" << endl;

    // Вводимо дані:
    cout << " x = ";
    cin >> x;    // вводимо змінну x
    cout << " eps = ";
    cin >> eps;    // вводимо точність

    // Підготовка до циклу
    member = x;    // перший член ряду дорівнює x
    S = member;    // перший доданок поміщаємо в S
    divider = 1;    // задаємо початкове значення дільника

    // Починаємо цикл
    while ( abs( member ) > eps )    // пам'ятаємо – ряд знакозмінний
    {
        divider += 2;    // перебираємо непарні значення
        // Знак члена ряду змінюємо унарно, домножуємо на
        // квадрат x та доділюємо на додатки до факторіалу:
        member = -member * x * x / ((divider - 1) * divider );
        S += member;    // підсумовуємо в S
    }
}

```


	<pre> n++; // збільшуємо кількість обертів циклу } cout << "\nСума числового ряду S = " << S << endl; // друкуємо cout << "Кількість оборотів циклу n = " << n << endl; // результат return 0; // успішне закінчення програми } // кінець функції main </pre>
☞	<pre> Програма обчислює суму елементів числового ряду з заданою точністю eps. x = 10.0 eps = 0.001 Сума числового ряду S = -0.544012 Кількість оборотів циклу n = 16 </pre>
☞	<pre> Програма обчислює суму елементів числового ряду з заданою точністю eps. x = 5.0 eps = 0.001 Сума числового ряду S = -0.958933 Кількість оборотів циклу n = 9 </pre>

Зробіть декілька розрахунків для різних значень змінної x ...

Пам'ятайте. Автор мови програмування C++ Б'ярне Страуструп сказав:
«Коли все зроблено добре, програмне забезпечення невидиме».

Лабораторна робота №5

Тема: Основи роботи одновимірними та двовимірними масивами в C++

Мета: освоєння роботи з масивами в C++.

Теоретична частина

5.1. Визначення одновимірних та багатовимірних масивів

Масив – це набір змінних однакового типу. Доступ до цих змінних здійснюється з допомогою одного й того ж імені. Це ім'я називається іменем масиву. Масиви використовуються для групування зв'язаних між собою змінних.

Масиви можуть бути одновимірними та багатовимірними. В одновимірних масивах для доступу до елементу масиву використовується один індекс. В багатовимірних масивах для доступу до елементу масиву використовується декілька індексів.

5.2. Опис одновимірного масиву

Загальна форма опису одновимірного масиву.

тип ім'яМасиву[розмір];

де тип – це тип елементів масиву. Він ще називається базовим типом. Базовий тип визначає розмір в байтах кожного елементу, що складає масив. Тип елементів масиву може бути як базовим типом так і складеним (структура);

ім'яМасиву – безпосередньо ім'я масиву, за яким здійснюється доступ до елементів масиву, що є його адресою в пам'яті комп'ютера;

розмір – кількість елементів в масиві.

Наприклад:

<pre>int a [5]; // масив з 5 елементів цілого типу double b [3]; // масив з 3 елементів дійсного типу</pre>
--

5.3. Способи опису масивів

По-перше, можна задавати кількість елементів масиву константами.

<pre>const int n = 3; // специфікатор const дає можливість програмісту // проінформувати компілятор, що значення даної // константи не повинно змінюватися в програмі double b [n]; // масив з n дійсних елементів</pre>

По-друге, масив можна ініціалізувати.

<pre>int a [] = {-5, 12, 10, -4}; // тут компілятор сам визначає // кількість елементів масиву: // масив з 4-х цілих елементів</pre>

По-третє, можна одночасно задавати число елементів масиву й ініціалізувати його.

```
double c [3] = {0.5, -6.33, 4.0}; // елементи масиву мають
                                // індекси: c[0], c[1], c[2]
```

Головне – потрібно запам'ятати, що звернення до елементів масиву проводиться за допомогою індексів. Перший індекс в масиві завжди дорівнює 0 (нулю)!

5.4. Ввод та вивід одновимірних масивів

Розглянемо та запрограмуємо рішення такої задачі: ввести в одновимірний масив й вивести п'ять цілих чисел.

```
➔ #include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
using namespace std;    // оголошуємо простір імен std

int main()    // функція main виконується першою
{
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // Описуємо змінні:
    const int n = 5; // кількість елементів масиву,
                    // кваліфікатор const, щоб не
                    // змінювали в програмі
    int A[n]; // масив з п'яти цілих чисел

    // Виводимо завдання:
    cout << "Ввести в одновимірний масив" << endl;
    cout << "й вивести п'ять цілих чисел. \n" << endl;


    // Вводимо масив:
    for ( int i = 0; i < n; i++ ) // пробігаємо елементи масива
    {
        cout << " A[" << i << "] = "; // запрошуємо ввести елемент
        cin >> A[i]; // вводимо елемент
    }

    cout << endl; // переводимо рядок

    // Виводимо масив:
    for ( int i = 0; i < n; i++ ) // пробігаємо елементи масива
        cout << " A[" << i << "] = " << A[i]; // виводимо масив

    cout << endl; // переводимо рядок

    return 0; // успішне закінчення програми
}            // кінець функції main
```



```

Ввести в одновимірний масив
й вивести п'ять цілих чисел.

A[0] = 12
A[1] = 346
A[2] = 756
A[3] = 95
A[4] = 876

A[0] = 12  A[1] = 346  A[2] = 756  A[3] = 95  A[4] = 876

```

5.5. Завдання до роботи з одновимірними масивами

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Завдання
1.	Дано два вектори $X = (x_1, x_2, \dots, x_{10})$ і $Y = (y_1, y_2, \dots, y_{10})$. Знайти скалярний добуток векторів і кут між ними. При обчисленні застосувати формулу $(x, y) = x \cdot y \cos \varphi$.
2.	Дано два вектори $X = (x_1, x_2, \dots, x_{15})$ і $Y = (y_1, y_2, \dots, y_{15})$. Знайти кут між ними та модуль векторного добутку векторів X і Y . При обчисленні застосувати формулу $ c = x \cdot y \sin \varphi$, де c - векторний добуток векторів X і Y .
3.	Дано два вектори $X = (x_1, x_2, \dots, x_7)$ і $Y = (y_1, y_2, \dots, y_7)$. Знайти суму векторів $X+Y$ і різницю векторів $X-Y$ та зберегти їх значення у векторах Z і F .
4.	Дано масив розміру N й цілі числа K и L ($1 \leq K \leq L \leq N$). Знайти суму елементів масива з номерами від K до L включно та їх середнє арифметичне.
5.	Дано два вектори $X = (x_1, x_2, \dots, x_{20})$ і $Y = (y_1, y_2, \dots, y_{20})$. Знайти добуток векторів X і Y на число α і зберегти їх результат у векторах G і H .

№	Завдання
6.	Дано два вектори $X = (x_1, x_2, \dots, x_{20})$ і $Y = (y_1, y_2, \dots, y_{20})$. Знайти лінійну комбінацію векторів $\alpha X + \beta Y$ і зберегти її у векторі U .
7.	Дано цілочисельний масив розміру N . Перевірити, чи чергуються в ньому парні і непарні числа. Якщо чергуються, то вивести 0, якщо ні, то вивести порядковий номер першого елемента, що порушує закономірність.
8.	Обчислити 25 значень функції $y = ax^2 + bx + c$ на відрізку $[e, f]$ та зберегти їх у масиві Y . Знайти мінімальне і максимальне значення в цьому масиві.
9.	Обчислити 12 значень функції $y = ax^2 + bx + c$ на відрізку $[e, f]$, зберегти їх у масиві Y і визначити, чи є функція спадною, зростаючою або немонотонною на цьому відрізку.
10.	Обчислити 18 значень функції $y = ax^2 + bx + c$ на відрізку $[e, f]$, зберегти їх у масиві Y і визначити, чи має рівняння $ax^2 + bx + c = 0$ на відрізку $[e, f]$ принаймні хоча б один корінь.
11.	Обчислити 30 значень функції $y = ax^2 + bx + c$ на відрізку $[e, f]$, зберегти їх у масиві Y . Негативні компоненти масиву Y скопіювати в масив $Y_{negative}$, а позитивні - у масив $Y_{positive}$.
12.	Обчислити 30 значень функції $y = ax^2 + bx + c$ на відрізку $[e, f]$, зберегти результати у масиві Y . Обчислити кількість негативних і позитивних компонентів масиву.
13.	У масиві $Time(10)$ зберігаються результати (у секундах) забігу на 100 метрів для 10 спортсменів. Визначити кращий результат і швидкість кожного спортсмена.
14.	У масиві X зберігаються дані про вартість (у гривнях) одного кілограма із семи основних продуктів харчування, а масиві Y – про добову потребу (у грамах) людини в цих продуктах. Визначити вартість добового раціону.
15.	У масиві $Kurs1(7)$ зберігаються дані про курс купівлі долара стосовно гривні для семи різних банків, а в масиві $Kurs2(7)$ – про курс продажу долара стосовно гривні для цих банків. Визначити найвигідніший курс обміну і банки, що працюють за цим курсом.
16.	Дано масив ненульових цілих чисел розміру N . Перевірити, чи чергуються в ньому позитивні і негативні числа. Якщо чергуються, то вивести 0, якщо немає, то вивести порядковий номер першого елемента, що порушує закономірність.

№	Завдання
17.	У масиві <i>Meat</i> (5) зберігаються дані про вартість (у гривнях) кілограма м'яса на п'яти різних ринках м. Дніпропетровська. Визначити середню, мінімальну і максимальну вартість м'яса. Вивести на екран “чорний список” ринків, де вартість м'яса вища середньої.
18.	У масиві <i>T</i> (5) зберігаються дані про температуру повітря на 1 січня 2018 р. у 5 містах світу. Визначити кількість міст, в яких температура була від'ємною, додатною і нульовою.
19.	Дано масив <i>A</i> розміру <i>N</i> . Знайти мінімальний елемент з його елементів з парними номерами: <i>A</i> ₂ , <i>A</i> ₄ , <i>A</i> ₆ , ..., а також максимальний елемент з його елементів з непарними номерами: <i>A</i> ₁ , <i>A</i> ₃ , <i>A</i> ₅ ,
20.	У масиві <i>Tair</i> (5) зберігаються відомості про середнє арифметичне значення температури води в Чорному морі за липень для 5 різних міст України, а в масиві <i>Twater</i> (5) – про середнє арифметичне значення температури повітря за липень в тих же містах. Визначити місто, в якому різниця між температурою повітря і води мінімальна.
21.	Дано масив розміру <i>N</i> . Знайти номери тих елементів масиву, які більше свого правого сусіда, і кількість таких елементів. Знайдені номери виводити в порядку їх зростання.
22.	Дано масив розміру <i>N</i> . Знайти номер його першого локального мінімуму (локальний мінімум – це елемент, який менше будь-якого зі своїх сусідів), а також номер його останнього локального максимуму (локальний максимум – це елемент, який більше будь-якого зі своїх сусідів).
23.	Дано масив розміру <i>N</i> . Знайти максимальний з його локальних мінімумів (локальний мінімум – це елемент, який менше будь-якого зі своїх сусідів), а також мінімальний з його локальних максимумів (локальний максимум – це елемент, який більше будь-якого зі своїх сусідів).
24.	Дано масив розміру <i>N</i> . Знайти кількість ділянок, на яких його елементи монотонно зростають.
25.	Дано масив розміру <i>N</i> . Знайти два сусідні елементи, сума яких максимальна, і вивести ці елементи в порядку зростання їх індексів.

5.6. Приклад розв'язання завдання до лабораторної роботи

	Записати 8 цілих чисел у масив <i>X</i> . Переписати позитивні з них в масив <i>P</i> , а негативні – в масив <i>N</i> .
--	--

Тут треба мати на увазі, що усі 8 цілих чисел можуть бути як позитивними, так і негативними, тому усі масиви описуємо на 8 елементів.



```
#include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
using namespace std;   // оголошуємо простір імен std

int main()    // функція main виконується першою
{
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // Описуємо змінні:
    const int n = 8;        // кількість елементів масиву,
                           // кваліфікатор const, щоб не
                           // змінювали в програмі
    int X[n],              // початковий масив
        P[n],              // масив позитивних чисел
        N[n],              // масив негативних чисел
        pos = 0,           // кількість позитивних елементів
        neg = 0;           // кількість негативних елементів

    // Виводимо завдання:
    cout << "Записати 8 цілих чисел у масив X. " << endl;
    cout << "Переписати позитивні з них в масив P," << endl;
    cout << "а негативні - в масив N.\n" << endl;

    // Вводимо масив X:
    for ( int i = 0; i < n; i++ ) // пробігаємо елементи масива
    {
        cout << " X[" << i << "] = "; // запрошуємо ввести елемент
        cin >> X[i];                  // вводимо елемент
    }

    cout << endl; // переводимо рядок

    // Заповнюємо масиви P та N:
    for ( int i = 0; i < n; i++ ) // пробігаємо елементи масива
    {
        if ( X[i] > 0 ) // перевіряємо на позитив
        {
            P[pos] = X[i]; // заносимо в масив для позитивних
            ++pos;         // підраховуємо кількість позитивних
        }
        if ( X[i] < 0 ) // перевіряємо на негатив
        {
            N[neg] = X[i]; // заносимо в масив для негативних
            ++neg;         // підраховуємо кількість негативних
        }
    }

    // Виводимо масив P:
    for ( int i = 0; i < pos; i++ ) // пробігаємо елементи масива
        cout << "  P[" << i << "] = " << P[i]; // виводимо масив
```

	<pre> cout << endl; // переводимо рядок // Виводимо масив N: for (int i = 0; i < neg; i++) // пробігаємо елементи масива cout << " N[" << i << "] = " << N[i]; // виводимо масив cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main </pre>
Ⓒ	<p>Записати 8 цілих чисел у масив X. Переписати позитивні з них в масив P, а негативні – в масив N.</p> <pre> X[0] = -23 X[1] = 45 X[2] = 67 X[3] = -82 X[4] = -65 X[5] = 2 X[6] = -5 X[7] = 456 P[0] = 45 P[1] = 67 P[2] = 2 P[3] = 456 N[0] = -23 N[1] = -82 N[2] = -65 N[3] = -5 </pre>

5.7. Опис двовимірного масиву

Загальна форма опису двовимірного масиву.

```
тип ім'яМасиву[розмір_1][розмір_2];
```

де тип – це тип елементів масиву. Він ще називається базовим типом. Базовий тип визначає розмір в байтах кожного елемента, що складає масив. Тип елементів масиву може бути як базовим типом так і складеним (структура);

ім'яМасиву – безпосередньо ім'я масиву, за яким здійснюється доступ до елементів масиву, що є його адресою в пам'яті комп'ютера;

розмір_1 – кількість так званих «рядків» в масиві за аналогією з матрицями;

розмір_2 – кількість так званих «стовпців» в масиві за аналогією з матрицями.

Наприклад:

```
int a[2][3];    // цілий масив з 2-х рядків і 3-х стовпців
double b[3][3]; // дійсний масив з 3-х рядків і 3-х стовпців
```

5.8. Способи опису двовимірних масивів

По-перше, можна задавати кількість елементів масиву константами.

```
const int n = 2; // число рядків масива
const int m = 3; // число стовпців масива
double b[n][m]   // двовірний масив дійсних чисел
```


По-друге, масив можна ініціалізувати, але другий індекс – обов'язковий.

```
double matr [[5] = {{3,5,5,7,8}, // ініціюємо перший рядок
                    {4,2,3,5,6}, // ініціюємо другий рядок
                    {7,8,9,7,3}}; // ініціюємо третій рядок
// тобто маємо масив:
double matr[3][5]; //перший елемент - matr[0][0] = 3
```

По-третє, можна одночасно задавати число елементів масиву й ініціалізувати його.

```
double matr [2][3] = {{2,8,5}, // ініціюємо перший рядок
                     {9,4,3}}; // ініціюємо другий рядок
// тобто маємо масив:
double matr[2][3]; // елементи масиву мають індекси:
// matr[0][0], matr[0][1], matr[0][2],
// matr[1][0], matr[1][1], matr[1][2],
```

5.9. Ввод та вивід двовимірних масивів

Розглянемо та запрограмуємо рішення такої задачі: ввести в двовимірний масив 3x3 шість дійсних чисел й вивести їх.

```
➔ #include <iostream> // підключаємо бібліотеку вводу-виводу
#include <windows.h> // підключаємо заголовний файл локалізації
using namespace std; // оголошуємо простір імен std

int main() // функція main виконується першою
{
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // Описуємо змінні:
    const int n = 3, // кількість рядків матриці
             m = 3; // кількість стовпців матриці
    double A[n][m]; // дійсний двовимірний масив

    // Виводимо завдання:
    cout << "Записати 6 дійсних чисел у двовимірний" << endl;
    cout << "масив й вивести їх.\n" << endl;

    // Вводимо масив A:
    for ( int i = 0; i < n; i++ ) // пробігаємо по рядкам
        for ( int j = 0; j < m; j++ ) // пробігаємо по стовпцям
        {
            cout << " A[" << i << "]["
                << j << "] = "; // запрошуємо ввести
елемент
            cin >> A[i][j]; // вводимо елемент
        }

    cout << endl; // переводимо рядок
```

	<pre> // Виводимо масив A: for (int i = 0; i < n; i++) // пробігаємо по рядкам for (int j = 0; j < m; j++) // пробігаємо по стовпцям cout << " A[" << i << "][" << j << "] = " << A[i][j]; // виводимо масив cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main </pre>
С	<p>Записати 6 дійсних чисел у двовимірний масив й вивести їх.</p> <pre> A[0][0] = 1.25 A[0][1] = 7.3 A[0][2] = 4.67 A[1][0] = 7.34 A[1][1] = 2.65 A[1][2] = 3.12 A[2][0] = 4.14 A[2][1] = 5.8 A[2][2] = 9.6 </pre> <p> A[0][0] = 1.25 A[0][1] = 7.3 A[0][2] = 4.67 A[1][0] = 7.34 A[1][1] = 2.65 A[1][2] = 3.12 A[2][0] = 4.14 A[2][1] = 5.8 A[2][2] = 9.6 </p>

5.10. Завдання до роботи з двовимірними масивами

Зауваження. Там, де не указано явно, вважати масив дійсним.

№	Завдання
1.	Дано масив розміром 5×6 з елементами цілого типу. Написати програму для обчислення кількості елементів масиву, кратних трьом, а також визначити суму $S(5)$ негативних елементів кожного рядка матриці.
2.	Дано масив розміром 7×5 з елементами дійсного типу. Написати програму для обчислення суми всіх додатних елементів масиву, а також визначити суму $S(5)$ позитивних елементів кожного стовпця матриці.
3.	Дано масив розміром 6×4 з елементами дійсного типу. Написати програму для обчислення кількості елементів масиву, відмінних від нуля, а також визначити добуток $P(6)$ позитивних елементів кожного рядка матриці.
4.	Дано масив розміром 4×6 з елементами цілого типу. Написати програму для обчислення суми парних елементів масиву, а також визначити добуток $P(6)$ непарних елементів стовпців матриці.
5.	Дано масив розміром 4×3 з елементами дійсного типу. Написати програму для обчислення добутку елементів масиву, модуль яких більше одиниці, а також визначити суму всіх позитивних елементів матриці та кількість нульових елементів

6.	Дано масив розміром 3×5 з елементами цілого типу. Написати програму для обчислення кількості елементів масиву, що при діленні на п'ять дають у залишку одиницю, а також визначити добуток всіх елементів матриці, попередньо замінивши нульові елементи на '-1'.
7.	Дано масив розміром 5×2 з елементами дійсного типу. Написати програму, що визначає наявність від'ємних елементів в масиві, а також визначити суму позитивних елементів, добуток негативних і кількість нульових елементів матриці.
8.	Дано масив розміром 9×4 з елементами дійсного типу. Написати програму для обчислення кількості елементів масиву, відмінних від '0', а також визначити мінімальний елемент матриці та його індекси.
9.	Дано масив розміром 3×7 з елементами дійсного типу. Написати програму, що визначає наявність у масиві хоча б одного елемента, що дорівнює нулю, а також визначити мінімальний і максимальний елементи матриці та їх добуток.
10.	Дано масив розміром 4×5 з елементами цілого типу. Написати програму для обчислення кількості елементів масиву, модуль яких менший 12, а також поміняти місцями мінімальний і максимальний елементи матриці.
11.	Дано масив розміром 2×5 з елементами дійсного типу. Написати програму для обчислення кількості елементів масиву, ціла частина яких дорівнює семи, а також визначити середнє арифметичне позитивних елементів матриці, добуток негативних елементів і кількість нульових.
12.	Дано масив розміром 6×4 з елементами дійсного типу. Написати програму для обчислення кількості елементів масиву, що при округленні дають шість, а також визначити суму $S(4)$ елементів, які перевищують по модулю одиницю, в кожному стовпці і кількість таких елементів.
13.	Дано масив розміром 3×5 з елементами цілого типу. Написати програму для обчислення суми квадратів елементів масиву, кратних трьом, а також визначити добуток $P(5)$ елементів, які перевищують $\pi/2$ в кожному стовпці та кількість таких елементів.
14.	Дано масив розміром 5×5 з елементами дійсного типу. Написати програму для обчислення номера рядка, в якому розміщено мінімальний елемент масиву, а також визначити суму квадратів елементів розташованих на обох діагоналях матриці.
15.	Дано масив розміром 5×5 з елементами цілого типу. Написати програму для обчислення номера стовпця, де міститься мінімальний елемент масиву, а також визначити суму позитивних елементів, розташованих над головною діагоналлю.
16.	Дано масив розміром 4×4 з елементами дійсного типу. Написати програму для обчислення кількості елементів масиву, дробова частина яких менша 0,1, а також визначити максимальний елемент на головній діагоналі та суму елементів головної діагоналі.

17.	Дано масив розміром 6×6 з елементами дійсного типу. Написати програму для обчислення добутку всіх позитивних елементів масиву, а також визначити суму негативних елементів розташованих під головною діагоналлю.
18.	Дано масив розміром 6×2 з елементами дійсного типу. Написати програму для обчислення різниці максимального за модулем елемента масиву та мінімального за модулем елемента масиву, а також визначити кількість позитивних елементів $P(6)$ у кожному рядку.
19.	Дано масив розміром 5×6 з елементами цілого типу. Написати програму для обчислення кількості позитивних елементів масиву, кратних п'яти, а також визначити кількість негативних елементів в усій матриці та в кожному стовпці $N(6)$.
20.	Дано масив розміром 7×2 з елементами цілого типу. Написати програму для обчислення суми всіх від'ємних елементів масиву, кратних двом, а також визначити кількість рядків, які містять нульові елементи, та їх номери.
21.	Дано масив розміром 3×5 з елементами цілого типу. Написати програму для обчислення кількості елементів масиву, що при діленні на шість дають у залишку одиницю, а також визначити для кожного стовпця: кількість ненульових елементів і їх добуток.
22.	Знайти середнє геометричне позитивних елементів кожного рядка матриці $Y(5 \times 6)$, а також для кожного рядка визначити кількість нульових елементів.
23.	Знайти найбільший елемент матриці $X(4 \times 4)$ і записати одиницю у той рядок і стовпець, в яких він міститься, а також визначити суми стовпців, які починаються з позитивного елемента та кількість негативних елементів матриці.
24.	Знайти найбільший елемент матриці $A(5 \times 3)$ і номер рядка та стовпця, в якому він міститься, а також визначити суму негативних елементів матриці, розташованих на перехресті непарних рядків і парних стовпців.
25.	Дано матрицю $A(6,5)$. Визначити суму елементів кожного рядка і суму елементів кожного стовпця, а також визначити суму елементів на головній діагоналі.

5.11. Приклад розв'язання завдання до лабораторної роботи

	Дано масив $A(3,4)$. Знайти найбільший елемент цього масиву та індекси цього елемента у масиві.
--	--

Програма для вирішення цієї задачі має наступний вигляд.

➔	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою {</pre>
---	---

```

SetConsoleOutputCP(1251); // локалізуємо вивід

// Описуємо змінні:
const int n = 3, // кількість рядків матриці
        m = 3; // кількість стовпців матриці
double A[n][m], // дійсний двовимірний масив
        max;    // значення максимального елемента
int iMax, // індекс макс. елемента в рядку
    jMax; // індекс макс. елемента в стовпці

// Виводимо завдання:
cout << "Дано матрицю A(3,4). Знайти найбільший елемент " <<
endl;
cout << "цього масиву та його індекси у масиві.\n" << endl;

// Вводимо масив A:
for ( int i = 0; i < n; i++ ) // пробігаємо по рядкам
    for ( int j = 0; j < m; j++ ) // пробігаємо по стовпцям
    {
        cout << " A[" << i << "]["
                << j << "] = "; // запрошуємо ввести
елемент
        cin >> A[i][j]; // вводимо елемент
    }

cout << endl; // переводимо рядок


// Шукаємо максимальний елемент і його індекси:
max = A[0][0]; // припускаємо, що 1-й елемент - максимальний
for ( int i = 0; i < n; i++ ) // пробігаємо по рядкам
    for ( int j = 0; j < m; j++ ) // пробігаємо по стовпцям
        if ( max < A[i][j] ) // порівнюємо max з наступним ел-
том
        {
            // якщо max менше наступного ел-
та
            max = A[i][j]; // оновлюємо max
            iMax = i; // зберігаємо індекс в рядку
            jMax = j; // зберігаємо індекс в стовпці
        }

cout << "Виводимо масив A:" << endl;
for ( int i = 0; i < n; i++ ) // пробігаємо по рядкам
    for ( int j = 0; j < m; j++ ) // пробігаємо по стовпцям
        cout << " A[" << i << "]["
                << j << "] = " << A[i][j]; // виводимо масив

cout << endl; // переводимо рядок

// Роздруковуємо результат:
cout << "\nМаксимальний елемент max = " << max << endl;
cout << "Індекс макс. ел-та у рядках iMax = " << iMax << endl;

```

	<pre> cout << "Індекс макс. ел-та у стовпцях jMax = " << jMax << endl; cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main </pre>
	<pre> Дано матрицю A(3,4). Знайти найбільший елемент цього масиву та його індекси у масиві. A[0][0] = 5.7 A[0][1] = 3.38 A[0][2] = 45.7 A[1][0] = 19.4 A[1][1] = 56.7 A[1][2] = 23.8 A[2][0] = 53.9 A[2][1] = 34.7 A[2][2] = 25.8 Виводимо масив A: A[0][0] = 5.7 A[0][1] = 3.38 A[0][2] = 45.7 A[1][0] = 19.4 A[1][1] = 56.7 A[1][2] = 23.8 A[2][0] = 53.9 A[2][1] = 34.7 A[2][2] = 25.8 Максимальний елемент max = 56.7 Індекс макс. ел-та у рядках iMax = 1 Індекс макс. ел-та у стовпцях jMax = 1 </pre>

Пам'ятайте. Відомий програміст Мартін Голдінг сказав: «Завжди пишіть код так, ніби супроводжувати його буде схильний до насильства психопат, який знає, де ви живете».

Лабораторна робота №6

Тема: Робота зі одновимірними та двовимірними динамічними масивами в C++

Мета: освоєння роботи з одновимірними та двовимірними динамічними масивами у мові C++. Їх використання для зберігання та обробки даних.

Теоретична частина

6.1. Визначення динамічних одновимірних та багатовимірних масивів

Динамічним називається масив, розмір якого може змінюватися під час виконання програми. Можливість зміни розміру відрізняє динамічний масив від статичного, розмір якого визначається на момент компіляції програми. Для виділення пам'яті для динамічного масиву та її звільнення мова програмування C++, надає відповідні вбудовані оператори. Динамічні масиви дають можливість більш гнучкої роботи з даними, оскільки дозволяють не прогнозувати обсяги даних, що будуть зберігатися, а регулювати розмір масиву відповідно до реально необхідних об'ємів.

Для створення динамічних масивів в C++ використовуються покажчики. Покажчик – це змінна, значенням якої є адреса комірки пам'яті, що зберігає дані певного типу. Зазвичай покажчик посилається на блок даних з області пам'яті, причому саме на його початок.

6.2. Робота з покажчиками і динамічними масивами

Покажчики різняться за типом в залежності від об'єктів, на які вони вказують. Під розміщення покажчиків всіх типів виділяється однаковий обсяг пам'яті, тому що розмір адреси залежить тільки від обчислювальної системи. Це зроблено для уникнення спроб присвоєння даних, що не відповідають типу покажчика, наприклад, змінної типу `double` за адресою покажчика `int` (тип `double` складається з 8-ми байт, а `int` – з 4-х!). А покажчики – це адреси комірок, з якими можна можна відтворювати арифметичні операції, наприклад, складення покажчика з цілою величиною. Тоді шаг при зміні адреси для `double` буде складатися з 8-ми байт, а `int` – з 4-х. І компілятор це відстежує.

Для роботи з адресами об'єктів в C++ використовуються дві основні операції:

1. `&` – операція взяття адреси об'єкта;
2. `*` – операція звернення за адресою об'єкта, на який вказує покажчик.

Наприклад:

➡	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int *yPtr; // оголошуємо покажчик на змінну цілого типу - // тут * вказує на тип змінної, // а yPtr - ім'я (адреса) змінної int y = 5; // оголошуємо статичну змінну цілого типу // зі значенням 5, тут y - ім'я (адреса) // змінної (ім'я для нас, а адреса // - для компілятора!) yPtr = &y; // заносимо адресу y в покажчик cout << "Адреса y = " << &y << endl; // виводимо адресу y cout << "Адреса y в yPtr = " << yPtr << endl; // виводимо адресу y cout << "Вміст y = " << *yPtr << endl; // виводимо вміст y = 5 cout << "\nЗаносимо 10 за адресою *yPtr:" << endl; *yPtr = 10; // тепер до y заносимо 10 cout << "Вміст y = " << y << endl; // виводимо вміст y = 10 delete yPtr; // звільняємо пам'ять cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main </pre>
Ⓢ	<pre> Адреса y = 0x28fef8 Адреса y в yPtr = 0x28fef8 Вміст y = 5 Заносимо 10 за адресою *yPtr: Вміст y = 10 </pre>

Динамічна пам'ять (ДП) виділяється операційною системою (ОС) за запитом програміста за допомогою спеціальних операторів C++. Це дозволяє в процесі роботи програми коригувати обсяг використовуваної пам'яті.

Динамічна пам'ять в мові C++ під відповідні типи даних (float, int, char та ін.) з використанням покажчиків виділяється за допомогою оператора new, А звільняється – за допомогою оператора delete.

Наприклад:

```

int *d = new int; // оголошуємо покажчик d на ціле число
                 // і виділяємо в пам'яті динамічну
                 // змінну, на яку він вказує

```



```
*d = 10;    // заносимо в динамічну змінну d число 10
cout << "*d = " << *d << endl; // виводимо вміст динамічної змінної
delete;    // обов'язково звільняємо пам'ять
```

6.3. Завдання до роботи з одновимірними динамічними масивами

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою Python з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

Зауваження. Кожен студент вибирає варіант завдання за номером у списку групи. Розмір початкового масиву $N \geq 10$ вводиться з клавіатури, а елементи розмістити в пам'яті динамічно. Дані в масив вводяться з клавіатури. Результати вивести на екран.

№	Завдання
1.	Дано масив розміру N і цілі числа K і L ($1 \leq K \leq L \leq N$). Знайти суму елементів масиву з номерами від K до L включно, а самі елементи динамічно розмістити в пам'яті і роздрукувати.
2.	Дано масив розміру N і цілі числа K і L ($1 \leq K \leq L \leq N$). Знайти середнє арифметичне елементів масиву з номерами від K до L включно, а елементи, що не входять в цей діапазон, динамічно розмістити в пам'яті і роздрукувати.
3.	Дано масив розміру N і цілі числа K і L ($1 < K \leq L \leq N$). Знайти суму всіх елементів масиву, крім елементів з номерами від K до L включно. Елементи з номерами від 1 до K динамічно розмістити в пам'яті і роздрукувати.
4.	Дано цілочисельний масив розміру N , в якому чергуються парні і непарні числа. Визначити кількість непарних чисел і розмістити їх динамічно в пам'яті і роздрукувати.
5.	Дано масив розміру N . Знайти мінімальний елемент з його елементів з парними номерами: A_2, A_4, A_6, \dots . Всі елементи масиву до мінімального елемента розмістити динамічно в пам'яті і роздрукувати.
6.	Дано масив A розміру N . Знайти мінімальний елемент з його елементів з парними номерами: A_2, A_4, A_6, \dots . Всі елементи масиву після мінімального елемента розмістити динамічно в пам'яті і роздрукувати.

7.	Дано масив розміру N . Знайти номери тих елементів масиву, які більше свого правого сусіда, і кількість таких елементів. Знайдені номери розмістити динамічно в пам'яті і роздрукувати.
8.	Дано масив розміру N . Знайти номери тих елементів масиву, які більше свого лівого сусіда, і кількість таких елементів. Знайдені номери розмістити динамічно в пам'яті і роздрукувати.
9.	Дано масив розміру N . Знайти номер його першого локального мінімуму (локальний мінімум – це елемент, який менше будь-якого зі своїх сусідів). Елементи, що знаходяться правіше локального мінімуму, розмістити в пам'яті динамічно і роздрукувати.
10.	Дано масив розміру N . Знайти номер його першого локального мінімуму (локальний мінімум – це елемент, який менше будь-якого зі своїх сусідів). Елементи, що знаходяться лівіше локального мінімуму, розмістити в пам'яті динамічно і роздрукувати.
11.	Дано масив розміру N . Знайти максимальний з його локальних мінімумів (локальний мінімум – це елемент, який менше будь-якого зі своїх сусідів). Елементи, що знаходяться правіше максимального локального мінімуму, розмістити в пам'яті динамічно і роздрукувати.
12.	Дано масив розміру N . Знайти максимальний з його локальних мінімумів (локальний мінімум – це елемент, який менше будь-якого зі своїх сусідів). Елементи, що знаходяться лівіше максимального локального мінімуму, розмістити в пам'яті динамічно і роздрукувати.
13.	Дано число R і масив A розміру N . Знайти елемент масиву, який найбільш близький до числа R (тобто такий елемент A_K , для якого величина $ A_K - R $ є мінімальною). Елементи, що знаходяться правіше A_K , розмістити в пам'яті динамічно і роздрукувати.
14.	Дано число R і масив A розміру N . Знайти елемент масиву, який найбільш близький до числа R (тобто такий елемент A_K , для якого величина $ A_K - R $ є мінімальною). Елементи, що знаходяться лівіше A_K , розмістити в пам'яті динамічно і роздрукувати.
15.	Дано масив розміру N . Знайти номери двох найближчих елементів з цього масиву (тобто елементів з найменшим модулем різниці) і вивести ці номери. Елементи між двома найближчими елементами розмістити в пам'яті динамічно і роздрукувати. Якщо елементи знаходяться поруч – видати відповідне повідомлення.
16.	Дано число R і масив A розміру N . Знайти два різних елемента масиву, сума яких найбільш близька до числа R , і вивести ці елементи, а елементи, які знаходяться між ними, розмістити в пам'яті динамічно і роздрукувати. Якщо елементи знаходяться поруч – видати відповідне повідомлення.
17.	Дано цілочисельний масив розміру N . Якщо він є <i>перестановкою</i> , тобто містить всі числа від 1 до N , то вивести 0, а в іншому випадку вивести номер першого неприпустимого елемента. Всі елементи до першого неприпустимого елемента розмістити в пам'яті динамічно і роздрукувати.

18.	Дано цілочисельний масив розміру N. Якщо він є <i>перестановкою</i> , тобто містить всі числа від 1 до N, то вивести 0, а в іншому випадку вивести номер першого неприпустимого елемента. Всі елементи після першого неприпустимого елемента розмістити в пам'яті динамічно і роздрукувати.
19.	Дано цілочисельний масив розміру N. Переписати в новий цілочисельний динамічно відведений масив B всі парні числа з вихідного масиву (в тому ж порядку) і вивести розмір отриманого масиву B і його вміст.
20.	Дано масив A розміру N. Сформувати новий динамічний масив B, в який записати всі позитивні елементи масиву A, зберігаючи вихідний порядок проходження елементів. Вивести розмір і вміст масиву B.
21.	Дано цілочисельний масив розміру N. Збільшити всі парні числа, що містяться в масиві, на початкове значення першого парного числа і розмістити в новому динамічному масиві. Вивести розмір і вміст масиву. Якщо парні числа в масиві відсутні, то залишити масив без змін і видати відповідне повідомлення.
22.	Дано масив A розміру N і цілі числа K і L ($1 \leq K < L \leq N$). Переставити в зворотному порядку елементи масиву, розташовані між елементами A_K і A_L , включаючи ці елементи, і розмістити в новому динамічному масиві. Вивести розмір і вміст масиву.
23.	Дано масив розміру N. Елементи масиву, розташовані між його мінімальним і максимальним елементами (не включаючи мінімальний і максимальний елементи) розмістити в новому динамічному масиві. Вивести розмір і вміст масиву.
24.	Дано масив розміру N. Переставити в зворотному порядку елементи масиву, розташовані між його мінімальним і максимальним елементами, включаючи мінімальний і максимальний елементи і розмістити в новому динамічному масиві. Вивести розмір і вміст масиву.
25.	Дано масив розміру N. Переставити в зворотному порядку елементи масиву, розташовані між його мінімальним і максимальним елементами, включаючи мінімальний і максимальний елементи і розмістити в новому динамічному масиві. Вивести розмір і вміст масиву.

6.4. Приклад розв'язання завдання до лабораторної роботи

	Заповнити одновимірний динамічний масив десятьма значеннями управляючої змінної циклу, а потім – новий динамічний одновимірний масив – останніми трьома з них.
--	--

Програма для вирішення цієї задачі має наступний вигляд.

➡	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід</pre>
---	---

```

// Описуємо змінну:
int n; // розмір одновимірного динамічного масиву

cout << "Введіть розмір масиву n = ";
cin >> n; // вводимо значення n

// Створюємо динамічний одновимірний масив:
int *pArr1 = new int[n]; // отримуємо покажчик на дин. масив
                        // і сам масив з n цілих елементів!!!

// Заповнюємо динамічний одновимірний масив:
for (int i = 0; i < n; i++) // використовуємо значення i
    pArr1[i] = i; // заповнюємо дин. масив значеннями i

cout << "\nРоздруковуємо масив pArr1:" << endl;
for (int i = 0; i < n; i++) // використовуємо значення i
    cout << "    pArr1[" << i
        << "] = " << pArr1[i]; // виводимо pArr1[i]

cout << endl; // переводимо рядок

// Створюємо новий динамічний одновимірний масив:
int *pArr2 = new int[3]; // отримуємо покажчик на дин. масив
                        // і сам масив з n цілих елементів!!!

// Заповнюємо масив pArr2:
for (int i = 0; i < 3; i++) // використовуємо значення i
    pArr2[i] = pArr1[i+7]; // заповнюємо з шагом 7

cout << "\nРоздруковуємо масив pArr2:" << endl;
for (int i = 0; i < 3; i++) // використовуємо значення i
    cout << "    pArr2[" << i
        << "] = " << pArr2[i]; // виводимо pArr2[i]

cout << endl; // переводимо рядок

// Після завершення всіх дій з дин. масивами:
delete[] pArr1; // обов'язково звільняємо динамічну пам'ять
delete[] pArr2; // обов'язково звільняємо динамічну пам'ять

return 0; // успішне закінчення програми
} // кінець функції main

```



```

Введіть розмір масиву n = 10
Роздруковуємо масив pArr1:
pArr1[0] = 0  pArr1[1] = 1  pArr1[2] = 2  pArr1[3] = 3  pArr1[4] = 4  pArr1[5]
= 5  pArr1[6] = 6  pArr1[7] = 7  pArr1[8] = 8  pArr1[9] = 9
Роздруковуємо масив pArr2:
pArr2[0] = 7  pArr2[1] = 8  pArr2[2] = 9

```

6.5. Робота з двовимірними динамічними масивами в C ++

У разі створення двовимірного динамічного масиву виділення пам'яті здійснюється наступним способом (рис. 5.1).

```
int nArr, // кількість рядків динамічного двовимірного масиву
    mArr; //кількість стовпців динамічного двовимірного масиву
cout << "Для динамічного двовимірного масиву" << endl
    << "введіть кількість рядків = ";
cin >> N; // вводимо кількість рядків
cout << "і стовпців = ";
cin >> M; // вводимо кількість стовпців
double **a = new float *[nArr]; // оголошуємо змінну типу «показчик
                                // на показчик на double»
                                // і виділяємо пам'ять
                                // під цей масив показників
// тепер «вказемо» кожному з показників отриманого масиву
// на масив типу double розмірності mArr:
for (int i = 0; i < nArr; i++) // організуємо цикл для виділення
                                // рядків масиву за кількістю
    a[i] = new float[mArr];    // стовпців, де кожен елемент
                                // масиву показників вказує на масив
                                // рядків, причому адреса кожного
                                // елемента - a[i][j]
```

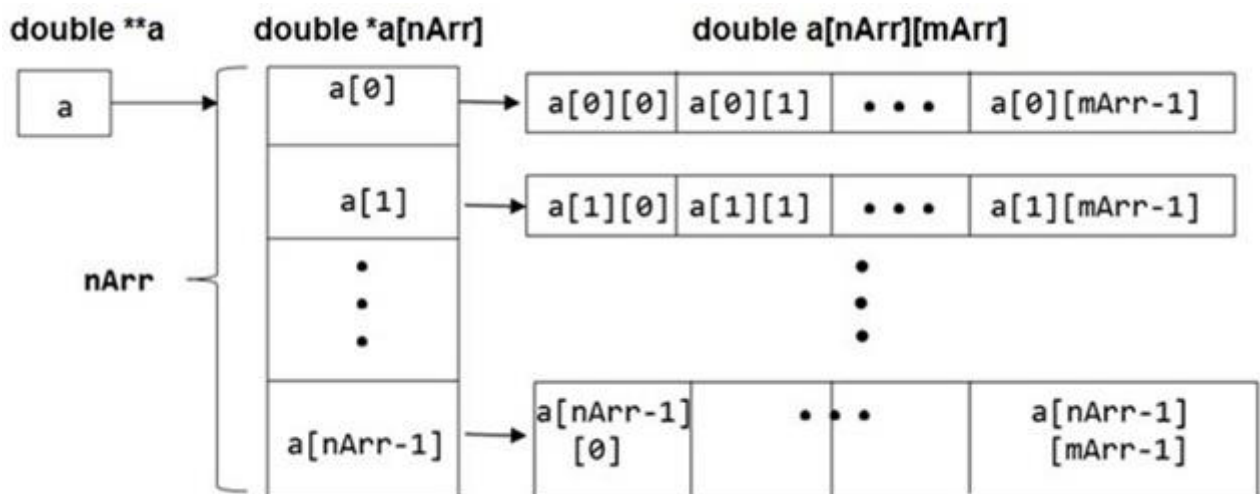


Рис. 5.1. Розташування компонентів в процесі створення двовимірного динамічного масиву

Видаляється такий масив у зворотному порядку.

```
for (int i=0; i < nArr; i++) // організовуємо цикл
    delete [] a[i]; // спочатку видаляємо масиви значень
delete [] a; // потім видаляємо масив показників
```

6.6. Завдання до роботи з двовимірними динамічними масивами

Зауваження. Кожен студент вибирає варіант завдання за номером у списку групи. Дані в матрицю вводяться з клавіатури. Результати вивести на екран.

№	Завдання
1.	Визначити добуток позитивних елементів кожного стовпця матриці, її розмір вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
2.	Визначити добуток елементів парних стовпців матриці, її розмір вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
3.	Визначити суму елементів матриці, які за модулем перевищують одиницю, у кожному парному стовпці і кількість таких елементів. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
4.	Визначити суму позитивних елементів розміщених над головною діагоналлю. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
5.	Визначити кількість рядків, які містять нульові елементи матриці, та їх номери. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
6.	Визначити значення максимального елемента кожного рядка матриці та номер стовпця, в якому він розміщений. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
7.	Максимальний і мінімальний елементи матриці поміняти місцями. Індеси зберегти у динамічно відведеному масиві. Розмір матриці вводити з клавіатури, а елементи розмістити в пам'яті динамічно.
8.	Визначити мінімальний і максимальний елементи матриці та їх добуток. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
9.	Визначити суму негативних елементів матриці, розташованих в кожному стовпчику під головною діагоналлю. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
10.	Визначити максимальний елемент матриці на головній діагоналі та суму елементів цієї діагоналі. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
11.	Визначити для кожного непарного рядка матриці кількість нульових елементів. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
12.	Знайти середнє геометричне позитивних елементів кожного рядка матриці, її розмір вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.

13.	Знайти найбільший елемент матриці та записати одиницю в той рядок і стовпець, на перехресті яких він міститься. Розмір матриці вводити з клавіатури, елементи, а також рядок та стовпець, розмістити в пам'яті динамічно.
14.	З матриці X побудувати матрицю Y , помінявши місцями рядки і стовпці. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
15.	Знайти найбільший елемент матриці і номери рядка і стовпчика, у яких він міститься. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
16.	Знайти найменший елемент матриці та записати в рядок, де міститься цей елемент, '10'. Розмір матриці вводити з клавіатури, елементи, а також рядок, де міститься '10', розмістити в пам'яті динамічно.
17.	Обчислити суму елементів кожного рядка матриці, визначити найменше значення цих сум і номер відповідного рядка. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
18.	Визначити кількість додатних і від'ємних елементів матриці. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
19.	Знайти найбільші елементи кожного рядка матриці X і записати їх у масив Y . Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
20.	Обчислити суму елементів кожного стовпця матриці, розташованих над головною діагоналлю. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
21.	Перемножити матриці $A(N \times M)$ і $B(N \times M)$. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
22.	Знайти найбільший елемент головної діагоналі матриці і зберегти в пам'яті увесь рядок, у якому він міститься. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
23.	Визначити максимальний елемент на головній діагоналі та добуток позитивних елементів зазначеної діагоналі. Розмір матриці вводити з клавіатури, елементи, а також результат, розмістити в пам'яті динамічно.
24.	Визначити добуток елементів парних стовпців матриці, її розмір вводити з клавіатури. Елементи матриці, а також результат, розмістити в пам'яті динамічно.
25.	Знайти суму елементів матриці, що мають задану різницю індексів $i-j=k$. Число k може бути і негативним. Розмір матриці та її елементи вводити з клавіатури і розмістити в пам'яті динамічно. Також, у динамічній пам'яті зберегти ті елементи матриці, що мають задану різницю індексів $i-j=k$.

6.7. Приклад розв'язання завдання до лабораторної роботи

	Дано динамічний масив дійсних чисел A розміру $N \times M$. сформувати одновимірний динамічний масив з елементів вихідного масиву, розташованих до максимального елемента.
--	---

Програма для вирішення цієї задачі має наступний вигляд.

```
➔ #include <iostream> // підключаємо бібліотеку вводу-виводу
#include <windows.h> // підключаємо заголовний файл локалізації

using namespace std; // оголошуємо простір імен std

int main() // функція main виконується першою
{
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // опис даних:
    int N, // кількість рядків динамічного двовимірного масиву
        M; // кількість стовпців динамічного двовимірного масиву
    cout << "Для динамічного двовимірного масиву" << endl
         << "введіть кількість рядків = ";
    cin >> N; // вводимо кількість рядків
    cout << "          і стовпців = ";
    cin >> M; // вводимо кількість стовпців
    cout << endl; // переводимо рядок

    float **A = new float *[N]; // оголошуємо змінну типу
    «покажчик                    // на покажчик на float» і
    виділяємо                    // пам'ять під цей масив
                                // показників

    // тепер «вказемо» кожним з показників отриманого масиву
    // на масив типу float розмірності M:
    for (int i=0; i<N; i++) // організуємо цикл для виділення
                           // рядків масиву по числу стовпців
        A[i] = new float [M]; // кожен елемент масиву показників
                           // вказує на масив рядків, причому
                           // адреса кожного елемента - A[i][j]

    // вводимо в масив A[N][M] числа з клавіатури
    cout << "Введіть в масив A[N][M] дійсні числа з клавіатури:" <<
    endl;
    for (int i=0; i<N; i++) // цикл по рядках
        for (int j = 0; j < M; j ++ ) // цикл по стовпцях
        {
            cout << "A[" << i << "][" << j // запрошуємо
                << "] = "; // до введення
            cin >> A[i][j]; // вводимо
        }
    cout << endl; // переводимо рядок

    // перевіряємо вміст масиву A[N][M]
    cout << "Дійсні числа в масиві A[N][M]:" << endl;
    for (int i=0; i<N; i++) // цикл по рядках
```



```

{
    for (int j=0; j<M; j++)          // цикл по стовпцях
        cout << "A[" << i << "]" << " " << j << "]" = " << A[i][j] // виводимо з індексами
        // по рядку i стовпцю
        << " ";                      // i доповнюємо двома
пробілами
        cout << endl; // кожний новий рядок починаємо з нового
рядка
    }
    cout << endl; // переводимо рядок

    // виконуємо завдання:
    // опис даних - відразу вважаємо, що:
    float max = A[0][0]; // значення максимального елемента
матриці A
    int iA = 0,          // індекс по рядку максимального елемента
        jA = 0,          // індекс по стовпцю максимального
елемента
        qnt = 0,         // число елементів матриці A до max
        k = 0;           // ще нам знадобиться індекс в новому дин.
мас.
        bool flag = 0;    // змінна для виходу з подвійного циклу

    // знаходимо max і кількість елементів до нього
    for (int i=0; i<N; i++) // цикл по рядках
        for (int j=0; j<M; j++) // цикл по стовпцях
            if ( A[i][j] > max ) // раптом новий елемент A більше
max
            {
                max = A[i][j]; // тоді це новий max
                iA = i;        // його індекс по рядку
                jA = j;        // його індекс по стовпцю
                ++qnt;         // і кількість ел-тів до max зросла
            }
        cout << endl; // переводимо рядок

    float *a = new float[qnt]; // отримуємо покажчик на дин. масив
                                // і сам масив з qnt вещ. елементів
    !!!

    cout << "Елементи одновимірною динамічного масиву:" << endl;
    for (int i=0; i<N; i++) // цикл по рядках дин. масиву A
    {
        for (int j=0; j<M; j++) // цикл по стовпцях дин.
масиву A
            if ( i == iA && j == jA ) // якщо дійшли до max
            {
                flag = 1; // піднімаємо прапор для виходу з циклу
по i
                break;    // виходимо з циклу по j
            }
    }

```

```

        else
        {
            a[k] = A[i][j]; // переносимо ел-ти матриці в дин.
мас.
            cout << "a[" << k << "] = " // одночасно
                << a[k] << " "; // роздруковуємо їх
            ++k; // просуваємо індекс в дин. масиві
        }
        if ( flag ) // якщо прапор піднято
            break; // виходимо з циклу по i
    }

    delete [] a; // обов'язково звільняємо динамічну пам'ять мас. a

    // двовимірний динамічний масив A видаляється в зворотному
порядку:
    for (int i=0; i<N; i++)
        delete [] A[i]; // спочатку видаляємо масиви значень
    delete [] A; // потім видаляємо масив покажчиків

    cout << endl; // переводимо рядок

    return 0; // успішне закінчення програми
} // кінець функції main

```



```

Для динамічного двовимірного масиву
введіть кількість рядків = 6
        і стовпців = 4

Введіть в масив A[N][M] дійсні числа з клавіатури:

A[0][0] = 1
A[0][1] = 2
A[0][2] = 3
A[0][3] = 4
A[1][0] = 5
A[1][1] = 6
A[1][2] = 7
A[1][3] = 8
A[2][0] = 9
A[2][1] = 100
A[2][2] = 1
A[2][3] = 2
A[3][0] = 3
A[3][1] = 4
A[3][2] = 5
A[3][3] = 6
A[4][0] = 7
A[4][1] = 8
A[4][2] = 9
A[4][3] = 1
A[5][0] = 2
A[5][1] = 3
A[5][2] = 4
A[5][3] = 5

```

Дійсні числа в масиві A[N][M]:

```
A[0][0] = 1  A[0][1] = 2  A[0][2] = 3  A[0][3] = 4  
A[1][0] = 5  A[1][1] = 6  A[1][2] = 7  A[1][3] = 8  
A[2][0] = 9  A[2][1] = 100 A[2][2] = 1  A[2][3] = 2  
A[3][0] = 3  A[3][1] = 4  A[3][2] = 5  A[3][3] = 6  
A[4][0] = 7  A[4][1] = 8  A[4][2] = 9  A[4][3] = 1  
A[5][0] = 2  A[5][1] = 3  A[5][2] = 4  A[5][3] = 5
```

Елементи одновимірного динамічного масиву:

```
a[0] = 1  a[1] = 2  a[2] = 3  a[3] = 4  a[4] = 5  a[5] = 6  a[6] = 7  a[7] = 8  
a[8] = 9
```

Пам'ятайте! Відомий програміст Гарольд Абелсон сказав: «Програми повинні писатися для людей, які будуть їх читати, а машини, які будуть ці програми виконувати – другорядні».

Лабораторна робота №7

Тема: Робота з функціями, які повертають і не повертають значення в C++

Мета: освоєння особливостей роботи з функціями, які повертають і не повертають значення в C++.

Теоретична частина

7.1. Визначення функцій

У комп'ютерному програмуванні функція – це послідовність операторів мови програмування, яка вирішує певну задачу і сформована як програмна одиниця (програмний модуль). У подальшому цей програмний модуль можна використовувати в різноманітних програмах, де б ця конкретна задача не виконувалася.

7.2. Опис функцій, які повертають значення

Щоб скористатися функцією у C++, вам слід виконати такі кроки:

- надати визначення функції;
- зобразити прототип функції;
- викликати функцію.

Загальна форма опису цих дій виглядає наступним чином.

```
. . .
тип ім'яФункції(списокТипівПараметрів)  // прототип функції
. . .
int main()  //
{
. . .
    тип змінна1,
        ...
        зміннаN;
. . .
    змінна1 = ім'яФункції(списокФактичнихПараметрів)  // виклик функції
    змінна2 = число * ім'яФункції(списокФактичнихПараметрів)  // функція
                                     // може приймати участь у арифметичних виразах
. . .
    return 0;
}

// визначення функції
тип ім'яФункції ( // списокФормальнихПараметрів функції:
```

```

    тип параметр1,    // ці параметри
. . .                // описують роботу
    тип параметрN    // функції з формальними!
){ // тіло функції
    тип зміннаВ1,    // ці внутрішні змінні
. . .                // служать для роботи
        зміннаВN;    // усередині функції

    Блок операторів,
    які виконуються,
    в тому числі -
    управляючих

    return вираз;    // функція повертає значення!
}

```

де **тип** – тип значення, яке повертає функція, а також змінних, які приймають участь у обчисленні функції;

ім'яФункції – це безпосередньо ім'я функції, яке повинно фігурувати в опису прототипа функції, виклику функції, а також опису функції;

списокТипівПараметрів – перелік типів змінних, які передаються у функцію;

списокФактичнихПараметрів – перелік змінних у виклику функції, які передаються функції для обчислень, причому, зверніть увагу на те, що виклики функцій, які повертають значення стоять *зліва від оператора присвоєння*;

списокФормальнихПараметрів – перелік змінних у опису функції, які формально описують хід обчислень у функції;

return – оператор повернення результату функцією.

Ви повинні добре розуміти, що функції, які повертають значення, для десяткових чисел становлять деяке *число*!

7.3. Використання функцій, які повертають значення

Розглянемо приклад обчислення гіпотенузи прямокутного трикутника за допомогою функції, яка повертає значення.


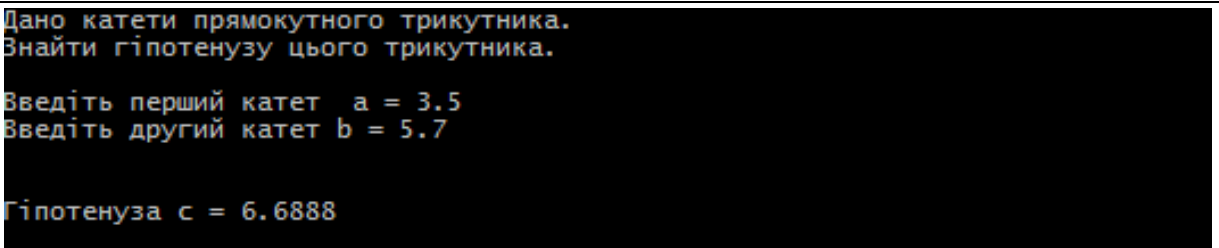
```

➡ #include <iostream>    // підключаємо бібліотеку вводу-виводу
   #include <cmath>      // підключаємо математичну бібліотеку
   #include <windows.h>   // підключаємо заголовний файл локалізації
   using namespace std;  // оголошуємо простір імен std

   double hypotenuseTriangle(double, double); // прототип функції

   int main() // функція main виконується першою
   {

```

	<pre> SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double a, // перший катет трикутника b, // другий катет трикутника c; // гіпотенуза трикутника // Виводимо завдання: cout << "Дано катети прямокутного трикутника." << endl; cout << "Знайти гіпотенузу цього трикутника.\n" << endl; cout << "Введіть перший катет a = "; cin >> a; // вводимо перший катет cout << "Введіть другий катет b = "; cin >> b; // вводимо перший катет cout << endl; // переводимо рядок c = hypotenuseTriangle(a, b); // обчислюємо гіпотенузу cout << "\nГіпотенуза c = " << c << endl; cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main // Функція знаходження гіпотенузи прямокутного трикутника: double hypotenuseTriangle (// опис формальних параметрів: double a, // перший катет прямокутного трикутника double b // другий катет прямокутного трикутника){ // Перевіряємо введені дані: if (a > 0 && b > 0) // якщо дані в порядку: return sqrt(b*b + a*a); // обчислюємо гіпотенузу else // інакше: cout << "\nПеревірте дані!" << endl; // попереджаємо! } </pre>
	

7.4. Завдання до роботи з функціями, які повертають значення

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти значення функції відповідно до варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

Зауваження. У другій гілці формули для y значення n для суми прийняти рівним п'яти.

№ п/п	Завдання
1.	$y = \begin{cases} 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \left(\frac{(i-1)!}{\cos ix} + \ln x \right), & \text{якщо } 2,0 \leq x < 5,5; \\ x^3 + e^{\cos^2 x^3}, & \text{інакше.} \end{cases}$
2.	$y = \begin{cases} 1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \dots \pm \frac{1}{n^2} \mp \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \left(\frac{i! + \cos x}{xi! + \operatorname{tg} ix} \right), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\operatorname{ctg}^2 x - \ln x }{\sqrt{ \cos x } + 4}, & \text{інакше.} \end{cases}$
3.	$y = \begin{cases} 1 + \frac{1}{4}x - \frac{1 \cdot 3}{4 \cdot 8}x^2 + \frac{1 \cdot 3 \cdot 7}{4 \cdot 8 \cdot 12}x^3 - \frac{1 \cdot 3 \cdot 7 \cdot 11}{4 \cdot 8 \cdot 12 \cdot 16}x^4 \pm \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \frac{\operatorname{tg} x (i+1)! + x}{(x+1)(i+1)!}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\operatorname{ctg} x}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$
4.	$y = \begin{cases} \frac{1}{2} + \frac{3}{2^2} + \dots + \frac{2n-1}{2^n} \mp \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (\cos^2 ix - \sin x^2)(i-1)!, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{2^x + (3+x)^x}{e^{ x-0,5 } + 3,6}, & \text{інакше.} \end{cases}$

№ п/п	Завдання
5.	$y = \begin{cases} \frac{\pi}{2} - \left(x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{2n+1} \pm \right) \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (\sqrt[i]{i!+x} - x^i), & \text{якщо } 1,0 \leq x < 5,5; \\ \frac{\lg x}{x^{3-\cos x}} + \frac{\arctg x + 2}{\sqrt{x+4}}, & \text{інакше.} \end{cases}$
6.	$y = \begin{cases} 1 + \frac{x \ln x}{1!} + \frac{(x \ln x)^2}{2!} + \dots + \frac{(x \ln x)^n}{n!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (-1)^i \frac{\sqrt[5]{i^2+x}}{e^i + x^2}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{ 1+x ^{\cos x} - 1,6}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$
7.	$y = \begin{cases} 1 + \frac{1}{x \cdot 3!} + \frac{1}{x^2 \cdot 4!} + \frac{1}{x^3 \cdot 5!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=2}^5 (-1)^i \frac{i^2}{i!-1} (\cos(xi!)), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{ctg^2 x - \ln x }{\sqrt{ \cos x } + 4}, & \text{інакше.} \end{cases}$
8.	$y = \begin{cases} x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (\sqrt[i]{i!+x} - x^i), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\lg x}{x^{3-\cos x}} + \frac{\arctg x + 2}{\sqrt{x+4}}, & \text{інакше.} \end{cases}$
9.	$y = \begin{cases} 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \left(\frac{i! + \cos x}{xi! + tgix} \right), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\arctg x}{ x + \frac{1}{\sqrt{x}}}, & \text{інакше.} \end{cases}$

№ п/п	Завдання
10.	$y = \begin{cases} x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (\sqrt{i!+x} - x^i), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\lg x}{x^{3-\cos x}} + \frac{\arctg x + 2}{\sqrt{x+4}}, & \text{інакше.} \end{cases}$
11.	$y = \begin{cases} x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n} \pm \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \frac{x}{x^{(i+1)!} + x} ((i-1)! - x), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\ln x^7 - e^{\frac{x}{2}}}{x^{3-\cos x}}, & \text{інакше.} \end{cases}$
12.	$y = \begin{cases} \frac{x-1}{x} + \frac{(x-1)^2}{2x^2} + \dots + \frac{(x-1)^n}{nx^n} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \frac{\lg x (i+1)! + x}{(x+1)(i+1)!}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\operatorname{ctg} x}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$
13.	$y = \begin{cases} x - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (-1)^{i-1} \frac{i! + x^2}{\lg^2 x^3 - xi!}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\sqrt[3]{8 + x-2 ^2}}{1 + \ln x}, & \text{інакше.} \end{cases}$
14.	$y = \begin{cases} x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \pm (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (\sqrt{i!+x} - x^i), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\lg x}{x^{3-\cos x}} + \frac{\arctg x + 2}{\sqrt{x+4}}, & \text{інакше.} \end{cases}$

№ п/п	Завдання
15.	$y = \begin{cases} \frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \dots + \frac{1}{(3n-2)(3n+1)} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (-1)^i \frac{\sqrt[5]{i^2 + x}}{e^i + x^2}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{ 1+x ^{\cos x} - 1,6}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$
16.	$y = \begin{cases} \frac{1}{1^4} + \frac{1}{3^4} + \dots + \frac{1}{(2n+1)^4} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \frac{x}{x^{(i+1)!} + x} ((i-1)! - x), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\ln x^7 - e^{\frac{x}{2}}}{x^{3-\cos x}}, & \text{інакше.} \end{cases}$
17.	$y = \begin{cases} 1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots \pm \frac{1}{n^4} \mp \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \left(\frac{i! + \cos x}{xi! + \operatorname{tg} ix} \right), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\operatorname{arctg} x}{ x + \frac{1}{\sqrt{x}}}, & \text{інакше.} \end{cases}$
18.	$y = \begin{cases} 1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots + \frac{1}{n^4} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n x^{(i-1)!} (\cos^2 x^5 + i^2), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\operatorname{ctg}^2 x - \ln x }{\sqrt{\cos x} + 4}, & \text{інакше.} \end{cases}$
19.	$y = \begin{cases} \frac{1}{1^2} + \frac{1}{3^2} + \dots + \frac{1}{(2n+1)^2} \mp \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (\cos^2 ix - \sin x^2) (i-1)!, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{2^x + (3+x)^x}{e^{ x-0,5 } + 3,6}, & \text{інакше.} \end{cases}$

№ п/п	Завдання
20.	$y = \begin{cases} \frac{1}{3 \cdot 5} + \frac{1}{7 \cdot 9} + \dots + \frac{1}{(4n-1)(4n+1)} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (-1)^i \frac{\sqrt[5]{i^2+x}}{e^i+x^2}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{ 1+x ^{\cos x} - 1,6}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$
21.	$y = \begin{cases} \frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \dots + \frac{1}{(2n-1)(2n+1)} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \frac{x}{x^{(i+1)!} + x} ((i-1)! - x), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\ln x^7 - e^{\frac{x}{2}}}{x^{3-\cos x}}, & \text{інакше.} \end{cases}$
22.	$y = \begin{cases} 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots \pm \frac{1}{2^n} \mp \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \left(\frac{i! + \cos x}{xi! + \operatorname{tg} ix} \right), & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\operatorname{arctg} x}{ x + \frac{1}{\sqrt{x}}}, & \text{інакше.} \end{cases}$
23.	$y = \begin{cases} x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (-1)^i \frac{\ln(i+1)! + \operatorname{tg} x}{x^3 + (i-1)!}, & \text{якщо } 2,0 \leq x < 5,5; \\ \frac{\lg x}{x^{3-\cos x}} + \frac{\operatorname{arctg} x + 2}{\sqrt{x+4}}, & \text{інакше.} \end{cases}$
24.	$y = \begin{cases} x + \frac{x^3}{2 \cdot 3} + \frac{1 \cdot 3 x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n x^i \frac{(i-1)!}{i^2 + \sin^2 x}, & \text{якщо } 2,0 \leq x < 5,0; \\ \frac{\operatorname{tg}^3 x - \ln x-1 }{\sqrt{\cos x + 10}}, & \text{інакше.} \end{cases}$

№ п/п	Завдання
25.	$y = \begin{cases} \frac{x-1}{x} + \frac{(x-1)^2}{2x^2} + \dots + \frac{(x-1)^n}{nx^n} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n \frac{\lg x (i+1)! + x}{(x+1)(i+1)!}, & \text{якщо } 2,0 \leq x < 5,0; \\ \frac{\operatorname{ctgx}}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$

7.5 Приклад розв'язання завдання до лабораторної роботи

	$y = \begin{cases} x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots, & \text{якщо } 1,0 \leq x < 2,0 \text{ та } \varepsilon = 0,0001; \\ \sum_{i=1}^n (-1)^i \frac{x + \lg i-x }{i! + \ln i-x }, & \text{якщо } 2,0 \leq x < 5,0; \\ \frac{\lg x}{x^3 + \cos x} + \frac{\operatorname{arctgx} + x}{x + \frac{x^4 - e^x}{\sin x}}, & \text{інакше.} \end{cases}$
--	--

У цьому завданні нам потрібно кожен гілку функції у зобразити у вигляді функції, яка повертає значення.

➔	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <cmath> // підключаємо математичну бібліотеку #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std double branch1(double, double); // прототип функції 1-ї гілки double branch2(double); // прототип функції 2-ї гілки double branch3(double); // прототип функції 3-ї гілки int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double x, // аргумент функції y, // значення функції eps; // точність обчислень // Виводимо завдання: cout << "Обчислити значення функції" << endl; cout << "з трьома гілками.\n" << endl; cout << "Введіть аргумент функції x = "; </pre>
---	--

```

cin >> x; // вводим аргумент функції

cout << endl; // переводимо рядок

// Обчислюємо функцію y:
if ( 1.0 <= x && x < 2.0 ) // перевіряємо умову 1-ї гілки
{
    cout << "Введіть точність eps = ";
    cin >> eps; // вводим точність
    y = branch1(x, eps); // обчислюємо 1-шу гілку
}
else
    if (2.0 <= x && x < 5.0 ) // перевіряємо умову 2-ї гілки
        y = branch2(x); // обчислюємо другу гілку
    else
        y = branch3(x); // обчислюємо третю гілку

cout << "\ny = " << y << endl; // виводимо значення функції y

cout << endl; // переводимо рядок

return 0; // успішне закінчення програми
} // кінець функції main

// Функція обчислення 1-ї гілки 1.0 <= x && x < 2.0:
double branch1 ( // опис формальних параметрів:
    double x, // значення аргументу функції
    double eps // значення точності обчислень
){
    // Описуємо внутрішні змінні:
    double member, // член ряду
        divider, // доділювач
        S; // сума


    member = x; // початкове значення члена ряду
    divider = 1.0; // початкове значення доділювача

    // Організуємо цикл за заданою точністю:
    while ( abs(member) > eps ) // ряд знакозмінний тому abs()
    {
        member = -member * x * x // обчислюємо
        черговий
        / ((divider + 1)*(divider +2)); // член ряду
        S += member; // обчислюємо суму
        divider += 2; // просуваємо доділювач
    }

    cout << "\nГілка 1!" << endl; // інформуємо!

    return S; // повертаємо значення
}

```

	<pre> } // Функція обчислення 2-ї гілки 2.0 <= x && x < 5.0: double branch2 (// опис формальних параметрів: double x // значення аргументу функції){ // Описуємо внутрішні змінні: double minus1 = 1, // початкове значення одиниці factI = 1, // початкове значення факторіалу i S = 0; // початкове значення суми // Організуємо цикл згідно формули: for (int i = 1; i <= 5; i++) // обчислюємо 5 членів ряду { minus1 = -minus1; // змінюємо знак одиниці factI *= i; // обчислюємо факторіал S += minus1 * (x + log10(x)) // обчислюємо суму / (factI + log(x)); } cout << "Гілка 2!" << endl; // інформуємо! return S; // повертаємо значення } // Функція обчислення 3-ї гілки: double branch3 (// опис формальних параметрів: double x // значення аргументу функції){ // Описуємо внутрішні змінні для спрощення: double part1, // перша частина функції part2; // друга частина функції // Обчислюємо частини функції: part1 = log10(x) / (pow(x, 3) + cos(x)); part2 = (atan(x) + x) / (x + (pow(x, 4) - exp(x)) / sin(x)); cout << "\nГілка 3!" << endl; // інформуємо! return part1 + part2; // повертаємо значення } </pre>
	<pre> Обчислити значення функції з трьома гілками. Введіть аргумент функції x = 1.5 Введіть точність eps = 0.0001 Гілка 1! y = -0.502505 </pre>

Гілка 2.

```
Обчислити значення функції  
з трьома гілками.  
  
Введіть аргумент функції  x = 2.7  
  
Гілка 2!  
  
y = -0.873203
```

Гілка 3.

```
Обчислити значення функції  
з трьома гілками.  
  
Введіть аргумент функції  x = 6.3  
  
Гілка 3!  
  
y = 0.00330984
```

Увага! У ході лабораторної роботи обчислити всі три гілки.

7.6. Опис функцій, які не повертають значення

Загальна форма опису цих функцій виглядає наступним чином.

```
. . .  
void ім'яФункції(списокТипівПараметрів)  // прототип функції  
. . .  
int main()  //  
{  
. . .  
    ім'яФункції(списокФактичнихПараметрів)  // виклик функції  
. . .  
    return 0;  
}  
  
// визначення функції  
void ім'яФункції ( // списокФормальнихПараметрів функції:  
    тип1 параметр1, // ці параметри  
    . . .           // описують роботу  
    типN параметрN // функції з формальними!  
) { // тіло функції  
    тип1 зміннаВ1; // ці внутрішні змінні  
    . . .           // служать для роботи  
    типN зміннаВN; // усередині функції
```

```
Блок операторів,  
які виконуються,  
в тому числі -  
управляючих
```

```
}
```

де **void** – пустий тип функції, який указує, що вона не використовує оператор `return`, а *приймає* вхідні об'єкти й *повертає результат* через *список параметрів*;

ім'яФункції – це безпосередньо ім'я функції, яке повинно фігурувати в опису прототипа функції, виклику функції, а також опису функції;

списокТипівПараметрів – перелік типів змінних, які передаються у функцію;

списокФактичнихПараметрів – перелік *об'єктів* у виклику функції, які передаються функції для обчислень, причому, зверніть увагу на те, що виклики функцій, які не повертають значення *займають окремий рядок* у функції `main`;

списокФормальнихПараметрів – перелік *об'єктів* у опису функції, які формально описують хід обчислень у функції, причому це можуть бути не тільки змінні, а й масиви, структури і т. ін.;

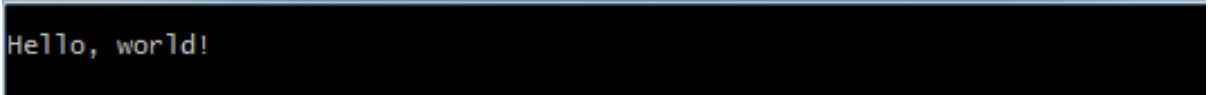
7.7. Використання функцій, які не повертають значення

У зв'язку з тим, що функції, які не повертають значення працюють через список параметрів, виникає ряд тонкощів при передачі різних об'єктів в ці функції.

7.7.1. Функції, які дійсно не повертають значення

На практиці іноді виникають випадки, коли деяка функція повинна просто виконати якісь дії.

```
➔ #include <iostream>    // підключаємо бібліотеку вводу-виводу  
   #include <windows.h>  // підключаємо заголовний файл локалізації  
   using namespace std;  // оголошуємо простір імен std  
  
   void hello(); // прототип функції 1-ї гілки  
  
   int main() // функція main виконується першою  
   {  
       SetConsoleOutputCP(1251); // локалізуємо вивід  
  
       hello(); // викликаємо функцію  
  
       return 0; // успішне закінчення програми  
   }             // кінець функції main
```


	<pre>// Функція : void hello (){ // Виконуємо задані дії: cout << "\nHello, world!\n" << endl; }</pre>
☞	

7.7.2. Передача у функцію змінних за значенням й за посиланням


У мові C++, якщо ви передаєте в функцію змінну *за значенням*, то функція діє не з нею, а з її копією. Щоб функція мала доступ до змінної у тілі головної функції `main`, потрібно передавати її *за посиланням* або *за покажчиком*. Тоді функція діє з *адресою змінної* з тіла головної функції `main` і має до неї доступ.

Тут важливим моментом є взаємодія програми, що запускається на виконання, з операційною системою (ОС). Основним завданням ОС є створення для неї процесу і виділення для нього *стека* і «*купи*».

«Купа» використовується для зберігання створюваних в процесі роботи програми динамічних змінних, а стек служить для організації доступу до більшості функцій.

Так, при виконанні кожної функції в стек заносяться її параметри в порядку «справа наліво». Тобто останній аргумент кладеться в стек в першу чергу, за ним кладеться передостанній аргумент, і так далі, поки всі аргументи не опиняться у стеці. Останнім в стек поміщається адреса повернення в головну функцію `main` – це адреса, що слідує за викликом функції в програмі. Після цього управління передається на початок функції. А функція виймає всі ці дані й працює з ними.

☞	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std // Прототипи функцій з передачею параметрів: void transByValue(int); // за значенням void transByReference(int &); // за посиланням void transByPointer(int *); // за покажчиком int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінну:</pre>
---	---

	<pre> int param = 50; cout << "\n Передаємо параметр за значенням:" << endl; transByValue(param); // викликаємо функцію cout << " transByValue: param = " << param << endl; cout << "\n Передаємо параметр за посиланням:" << endl; transByReference(param); cout << " transByReference: param = " << param << endl; cout << "\n Передаємо параметр за за покажчиком:" << endl; transByPointer(&param); cout << " transByPointer: param = " << param << endl; return 0; // успішне закінчення програми } // кінець функції main // Функція приймає параметр за значенням: void transByValue (// опис формальних параметрів: int param // копія параметру){ param = 100; // присвоюємо 100 } // Функція приймає параметр за посиланням: void transByReference (// опис формальних параметрів: int &param // значення адреси параметра){ param = 200; // присвоюємо 200 } // Функція приймає параметр за покажчиком: void transByPointer (// опис формальних параметрів: int *param // значення адреси параметра){ *param = 300; // присвоюємо 300 } </pre>
	<pre> Передаємо параметр за значенням: transByValue: param = 50 Передаємо параметр за посиланням: transByReference: param = 200 Передаємо параметр за за покажчиком: transByPointer: param = 300 </pre>

Тому, коли вам потрібно передати адресу параметра у функцію, необхідно дотримуватися необхідних правил при опису параметрів у прототипах, у виклику функції (за покажчиком) й у опису функцій.

7.7.3. Передача одновимірних і двовимірних масивів у функції, які не повертають значення

У C++ імена одновимірного та двовимірного масиву при переданні у функцію трактуються як його адреса, тому відповідний формальний параметр у опису функції є покажчиком. Тому функції, що викликаються, можуть модифікувати значення елементів у вихідних масивах викликальних функцій. Значенням назви масиву є адреса першого елемента масиву в пам'яті комп'ютера. Оскільки передається адреса початку масиву, викликана функція точно знає, де в пам'яті зберігається масив. Таким чином, коли викликана функція модифікує у своєму тілі елементи переданого масиву, вона модифікує дійсні елементи масиву в їх вихідних комірках пам'яті.

Розглянемо особливості формування прототипів функцій у які передаються одновимірні та двовимірні масиви.

```
. . .
const int n = 3;  // кількість рядків матриці
const int m = 4;  // кількість стовпців матриці

// Прототипи функцій
void func1(double[], int); // функція для обробки одновимірного масиву:
    // квадратні дужки вказують на те, що це одновимірний масив,
    // причому, розмір масива можна не указувати!
void func2(int[][m], int, int); // функція для обробки двовимірного
    // масиву: дві пари квадратних дужок вказують на те, що це
    // двовимірний масив, причому обов'язково потрібно указувати
    // кількість стовпців у других квадратних дужках!

. . .
int main() // функція main виконується першою
{
    . . .
    //
    // Описуємо змінні:
    double arr1[n]; // одновимірний масив з n елементів
    int arr2[n][m]; // двовимірний масив розміром nxm
    . . .
    func1(arr1, n); // викликаємо першу функцію
    func2(arr2, n, m); // викликаємо другу функцію
    . . .
    return 0;
}

// Функція обробки одновимірного масива:
void func1 ( // опис формальних параметрів:
```

```

    double arr1[], // одновимірний масив для обробки
    int n          // розмір масива
){
    . . . // тіло функції
}

// Функція обробки двовимірного масиву:
void func2 ( // опис формальних параметрів:
    int arr2[][m], // двовимірний масив для обробки
    int n,         // кількість
    int m          //
){
    . . . // тіло функції
}

```

7.8. Завдання до роботи з функціями, які не повертають значення

№	Завдання
1.	Дано матриці $A(3,3)$, $B(4,4)$. Для кожної з них обчислити суму елементів над, під головною діагоналлю та на головній діагоналі. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці та обчислення сум оформити у вигляді функцій.
2.	Дано матриці $A(2,4)$, $B(5,2)$, $C(4,4)$. Для кожної з них знайти найбільший і найменший елементи головної діагоналі. Результати передати через параметри функції й роздрукувати.. Уведення, виведення матриці, пошук мінімального і максимального елементів оформити у вигляді функцій.
3.	Дано матриці $A(2,2)$, $B(3,3)$, $C(4,4)$. Для кожної з них визначити максимальні елементи у рядках і перемістити їх на головну діагональ. Також визначити у кожній матриці кількість нульових елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці та її перетворення оформити у вигляді функцій.
4.	Дано матриці $A(2,3)$, $B(3,4)$, $C(4,4)$. Знайти середні геометричні позитивних елементів кожного їх рядка. Також знайти різницю абсолютних значень сум позитивних та негативних елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці та пошук середніх геометричних оформити у вигляді функцій.
5.	Дано матриці $A(3,3)$, $B(4,2)$, $C(2,5)$. Знайти найбільший елемент кожної з них і записати одиницю у той рядок і стовпець, в яких він розмішений. Також знайти суму добутків елементів з парними та з непарними індексами. Результати передати через параметри функції й роздрукувати. Уведення, виведення і перетворення матриці оформити у вигляді функцій.

6.	Дано матриці $A(2,2)$, $B(3,3)$, $C(4,4)$. Транспонувати їх, а потім знайти в кожній максимальний по абсолютному значенню елемент. Результати передати через параметри функції й роздрукувати. Уведення, виведення і транспонування матриці оформити у вигляді функцій.
7.	Дано матриці $A(3,2)$, $B(4,3)$, $C(3,4)$. Знайти найбільший елемент кожної з них і номер рядка і стовпця, в яких він розміщений, а також квадратний корінь з добутка квадратів значень елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці і пошук максимального елемента оформити у вигляді функцій.
8.	Дано матриці $A(4,3)$, $B(3,4)$, $C(5,2)$. Знайти найменший елемент кожної з них і записати у рядок, де міститься цей елемент, '10'. Також знайти середнє арифметичне значення абсолютних величин елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці та перетворення матриці оформити у вигляді функцій.
9.	Дано матриці $A(4,3)$, $B(2,5)$, $C(3,4)$. Для кожної з них обчислити суму елементів рядків і визначити найменше значення цих сум і номер відповідного рядка. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці і пошук найменшого значення з сум рядків та номера рядка оформити у вигляді функцій.
10.	Дано матриці $A(3,3)$, $B(4,3)$, $C(2,4)$. Визначити кількість додатних та від'ємних елементів, а також різницю між добутком та сумою елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці і пошук кількості додатних та від'ємних елементів, а також різниці між добутком та сумою елементів оформити у вигляді функцій.
11.	Дано матриці $A(3,2)$, $B(3,3)$, $C(4,2)$. Для кожної з них знайти найбільші елементи кожного рядка і записати в одновимірний масив, а також знайти різницю сум елементів з парними та з непарними індексами. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, формування одновимірного масиву та знаходження різниці сум елементів з парними та з непарними індексами оформити у вигляді функцій.
12.	Дано матриці $A(4,3)$, $B(3,4)$ і два дійсних числа x і y . Знайти $x*A$, $y*B$, $x*C+y*B$. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, множення на число та складання масивів оформити у вигляді функцій.
13.	Дано матриці $A(2,2)$, $B(3,3)$, $C(4,4)$. Знайти транспоновані матриці A^m , B^m , C^m та кількість нульових елементів у них. Результати передати через параметри функції й роздрукувати. Уведення, виведення, транспонування матриці та знаходження кількості нульових елементів оформити у вигляді функцій.

14.	Дано матриці $A(2,2)$, $B(3,3)$, $C(4,4)$. Для кожної з них обчислити суму елементів над, під головною діагоналлю та на головній діагоналі. Знайти максимум з цих сум. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці та обчислення сум оформити у вигляді функцій.
15.	Дано матриці $A(2,2)$, $B(3,3)$, $C(4,4)$. Знайти третю норму для кожної з них та індекс i максимального елемента. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, обчислення норми та обчислення індекса i максимального елемента оформити у вигляді функцій, скориставшись формулою: $\ A\ _{III} = \sqrt{\sum_{i=1}^n \sum_{k=1}^n a_{ik}^2}.$
16.	Дано матриці $A(3,4)$, $B(5,2)$, $C(4,2)$, у кожній з них поміняти місцями перший та останій рядки, а також обчислити добуток елементів масива, які перебільшують значення $D1$ (для масива A), $D2$ (для масива B), $D3$ (для масива C). Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, обмін рядків та обчислення добутку елементів масива, які перебільшують значення $D1$ (для масива A), $D2$ (для масива B), $D3$ (для масива C) оформити у вигляді функцій.
17.	Дано матриці $A(4,3)$, $B(5,2)$, $C(4,2)$, у кожній з яких поміняти місцями перший і останній стовпці, а також знайти суму елементів з парними індексами. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, обмін стовпців і знаходження суми елементів з парними індексами оформити у вигляді функцій.
18.	Дано матриці $A(3,4)$, $B(5,2)$, $C(4,2)$, у кожній з яких визначити місце розміщення максимального елемента, а також обчислити різницю сум елементів з парними та з непарними індексами. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, пошук місця та обчислення різниці сум елементів з парними та з непарними індексами оформити у вигляді функцій.
19.	Дано матриці $A(4,3)$, $B(5,2)$, $C(4,2)$. Утворити три нові матриці діленням кожного елемента вихідної матриці на середнє арифметичне її елементів, а також обчислити різницю абсолютних значень сум позитивних та негативних елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення, зміну матриць та обчислення різниці абсолютних значень сум позитивних та негативних елементів оформити у вигляді функцій.
20.	Дано матриці $A(3,3)$, $B(5,2)$, $C(4,3)$, у кожній з яких поміняти місцями елементи на головній і побічній діагоналях, а також обчислити середнє арифметичне значення елементів, які перебільшують значення $P1$ (для масива A), $P2$ (для масива B), $P3$ (для масива C). Результати передати через параметри функції й роздрукувати. Уведення, виведення, перетворення матриці, а також обчислення середнього арифметичного значення елементів, які перебільшують значення $P1$ (для масива A), $P2$ (для масива B), $P3$ (для масива C) оформити у вигляді функцій.

21.	Дано матриці $A(3,3)$, $B(5,2)$, $C(4,4)$. Для кожної з них переставити два рядка матриці: $l1$ – рядок, який переставляється з рядком $l2$, а також обчислити різницю абсолютних значень сум позитивних та негативних елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, перестановку рядків та обчислення різниці абсолютних значень сум позитивних та негативних елементів оформити у вигляді функцій.
22.	Дано матриці $A(3,5)$, $B(5,2)$, $C(4,4)$. Для кожної з них обчислити суми рядків та різницю між добутком та сумою елементів. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, обчислення сум рядків, виведення вектора сум та обчислення різниці між добутком та сумою елементів оформити у вигляді функцій.
23.	Дано матриці $A(4,5)$, $B(5,3)$, $C(2,4)$. Для кожної з них обчислити суми стовпців та суму елементів, для яких виконується відповідна умова: $x_1 \leq a_i \leq y_1$; $x_2 \leq b_i \leq y_2$; $x_3 \leq c_i \leq y_3$. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, обчислення сум стовпців, виведення вектора сум та обчислення сум відповідних елементів оформити у вигляді функцій.
24.	Дано матриці $A(3,3)$, $B(5,5)$, $C(4,4)$. Для кожної з них знайти максимальний і мінімальний елементи головної діагоналі, а також підсумувати елементи кожного рядка матриці. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, пошук мінімального та максимального елементів, підсумовування елементів кожного рядка матриці, виведення вектора сум оформити у вигляді функцій.
25.	Дано матриці $A(3,5)$, $B(5,2)$, $C(4,5)$. Для кожної з них обчислити кількість додатних і від'ємних елементів, а також розділити кожну матрицю на дві матриці по рядку, де l – рядок, по якому проходить розподіл матриці (він належить нижній частині матриці). Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, обчислення кількості названих елементів та розділення матриць оформити у вигляді функцій.

7.9. Приклад розв'язання завдання до лабораторної роботи

Дано матриці $A(4,3)$ та $B(3,4)$. Для кожної з них знайти найбільший елемент та вивести його, а також записати 1 на те місце, де він знаходиться. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, знаходження найбільшого елементу й запис на його місце 1 оформити у вигляді функцій.

У цьому завданні нам потрібно застосовувати свої знання щодо розробки програм з використанням функцій, які не повертають значення та обробки масивів.

```
➔ #include <iostream>    // підключаємо бібліотеку вводу-виводу
#include <windows.h>    // підключаємо заголовний файл локалізації
using namespace std;    // оголошуємо простір імен std

//
const int n = 4;    // кількість рядків з двох матриць max = 4
const int m = 4;    // кількість стовпців з двох матриць max = 4

// Прототипи функцій:
void enterArray(double [][][m], int, int, char);    // ввід масива
void printArray(double [][][m], int, int, char);    // вивід масива
void changeArray(double [][][m], int, int, double &);    // обробка

int main()    // функція main виконується першою
{
    SetConsoleOutputCP(1251);    // локалізуємо вивід

    // Описуємо змінні:
    double array [n][m],    // у цьому масиві можна обробляти
                           // обидва масиви A(4,3) та B(3,4)
        maxElm;    // максимальний елемент масива

    // Виводимо завдання:
    cout << " Дано матриці A(4,3) та B(3,4). Для кожної з \n";
    cout << " них знайти найбільший елемент та вивести \n";
    cout << " його, а також записати 1 на те місце, де він \n";
    cout << " знаходиться. Результати передати через \n";
    cout << " параметри функції й роздрукувати. Уведення, \n";
    cout << " виведення матриці, знаходження найбільшого \n";
    cout << " елементу й запис на його місце 1 оформити \n";
    cout << " у вигляді функцій. \n";

    // Працюємо з матрицею A(4,3):
    cout << "\n Введіть масив A(4,3):" << endl;
    enterArray(array, n, 3, 'A');    // вводимо масив
    changeArray(array, n, 3, maxElm);    // обробляємо його
    printArray(array, n, 3, 'A');    // виводимо масив
    cout << "\n Макс. елемент масива A(4,3) = " << maxElm << endl;

    cout << endl;    //

    // Працюємо з матрицею B(3,4):
    cout << "\n Введіть масив B(3,4):" << endl;
    enterArray(array, 3, m, 'B');    // вводимо масив
    changeArray(array, 3, m, maxElm);    // обробляємо його
```



```

printArray(array, 3, m, 'B');    // виводимо масив
cout << "\n Макс. елемент масива B(3,4) = " << maxElm << endl;

cout << endl; //

return 0; // успішне закінчення програми
}          // кінець функції main


// Функція вводу масива:
void enterArray ( // опис формальних параметрів:
    double array[][m], // масив для вводу
    int n,              // кількість рядків масиву
    int m,              // кількість стовпців масиву
    char ch             // ім'я масива
){
    // Проходимо всі елементи масиву:
    for (int i = 0; i < n; i++)          // цикл по рядкам
        for (int j = 0; j < m; j++)      // цикл по стовпцям
        {
            cout << ch << "[" << i << "]"[" // виводимо
                << j << "] = ";          // запрошення
            cin >> array[i][j];           // вводимо елемент
        }
}

// Функція виводу масива:
void printArray ( // опис формальних параметрів:
    double array[][m], // масив для виводу
    int n,              // кількість рядків масиву
    int m,              // кількість стовпців масиву
    char ch             // ім'я масива
){
    // Проходимо всі елементи масиву:
    for (int i = 0; i < n; i++)          // цикл по рядкам
        for (int j = 0; j < m; j++)      // цикл по стовпцям
            cout << " " << ch << "[" << i // виводимо
                << "]"[" << j << "] = "   // елементи
                << array[i][j];           // построково

    cout << endl; // переводимо рядок
}

// Функція пошуку максимального елемента:
void changeArray ( // опис формальних параметрів:
    double array[][m], // оброблюваний масив
    int n,              // кількість рядків масиву
    int m,              // кількість стовпців масиву
    double &maxElm      // максимальний елемент
){
    // Описуємо внутрішні змінні:
    int iMax = 0, // індекс по рядку макс. елемента

```

	<pre> jMax = 0; // індекс по стовпцю макс. елемента // призначаємо індекси першого елемента maxElm = array[0][0]; // вважаємо перший елемент максимальним // Проходимо всі елементи масиву: for (int i = 0; i < n; i++) // цикл по рядкам for (int j = 0; j < m; j++) // цикл по стовпцям if (maxElm < array[i][j]) // перевіряємо, чи не є { // наступний ел-т максимальним maxElm = array[i][j]; // тоді заносимо у maxElm iMax = i; // перевизначаємо iMax jMax = j; // перевизначаємо jMax } // Заносимо одиницю у рядок iMax: for (int j = 0; j < m; j++) // цикл по стовпцям array[iMax][j] = 1.0; // заносимо одиницю // Заносимо одиницю у стовпець jMax: for (int i = 0; i < n; i++) // цикл по рядкам array[i][jMax] = 1.0; // заносимо одиницю } </pre>
	<p>Дано матриці A(4,3) та B(3,4). Для кожної з них знайти найбільший елемент та вивести його, а також записати 1 на те місце, де він знаходиться. Результати передати через параметри функції й роздрукувати. Уведення, виведення матриці, знаходження найбільшого елемента й запис на його місце 1 оформити у вигляді функцій.</p> <p>Введіть масив A(4,3):</p> <pre> A[0][0] = 12.6 A[0][1] = 4.5 A[0][2] = 38.2 A[1][0] = 16.3 A[1][1] = 65.5 A[1][2] = 22.7 A[2][0] = 3.6 A[2][1] = 46.4 A[2][2] = 7.2 A[3][0] = 83.7 A[3][1] = 55.3 A[3][2] = 61.8 </pre> <pre> A[0][0] = 1 A[0][1] = 4.5 A[0][2] = 38.2 A[1][0] = 1 A[1][1] = 65.5 A[1][2] = 22.7 A[2][0] = 1 A[2][1] = 46.4 A[2][2] = 7.2 A[3][0] = 1 A[3][1] = 1 A[3][2] = 1 </pre> <p>Макс. елемент масива A(4,3) = 83.7</p>

Введіть масив B(3,4):

B[0][0] = 4.5

B[0][1] = 7.3

B[0][2] = 43.7

B[0][3] = 14.6

B[1][0] = 25.8

B[1][1] = 33.7

B[1][2] = 5.8

B[1][3] = 63.7

B[2][0] = 16.9

B[2][1] = 52.8

B[2][2] = 4.3

B[2][3] = 73.6

B[0][0] = 4.5 B[0][1] = 7.3 B[0][2] = 43.7 B[0][3] = 1 B[1][0] = 25.8 B[1]

[1] = 33.7 B[1][2] = 5.8 B[1][3] = 1 B[2][0] = 1 B[2][1] = 1 B[2][2] = 1

B[2][3] = 1

Макс. елемент масива B(3,4) = 73.6

Пам'ятайте. Генрі Луїс Менкен сказав: «Для кожного складного завдання існує рішення, яке є швидким, простим і неправильним».

Лабораторна робота №8

Тема: Основи роботи з символами та рядками в C++

Мета: освоєння роботи з символами та рядками в C++.

Теоретична частина

8.1. Визначення символів та рядків в C++

Рядок – послідовність (масив) символів. Якщо ми діємо з окремим *символом*, то його треба брати в *одинарні лапки*. Символи відносяться до типу `char`. При використанні набору символів у вигляді рядка останній має бути у *подвійних лапках*. Хоча тип `char` і відноситься до цілочисельних типів даних (і, таким чином, слідує всім їх правилам), робота з `char` дещо відрізняється від роботи зі звичайними цілочисельними типами.

У стандарт C++ включено підтримку декількох наборів символів. Традиційний 8-бітовий набір символів ASCII (American Standard Code for Information Interchange – Американський стандартний код для обміну інформацією) називається «вузькими» символами, а 16-бітовий – «широкими». Для кожного з них у бібліотеках C++ `<cstring>` та `<string.h>` є свої набори функцій. ASCII – визначає спосіб представлення символів англійської мови (+ кілька інших) у вигляді чисел від 0 до 127 (див. Додаток 3). Коди від 0 по 32 називаються управляючими символами. Коди з 33 по 64 відповідають знакам арифметичних операцій, службовим символам та цифрам. Коди з 65 по 90 відповідають великим буквам англійського алфавіту, а з 97 по 122 – маленьким.

8.2. Робота з символами та рядками в C++

Для зображення символічних рядків у C++ не існує спеціального рядкового типу! Вони представляються як масиви елементів типу `char` і закінчуються «термінатором рядка» – символом з нульовим значенням виду `'\0'` і тому називаються ASCII-рядки. Як ми вже казали, рядки беруть у *подвійні лапки*, а символи – в *одинарні*.

У C++ тип `char` відноситься до цілочисельних типів даних, але робота з `char` відрізняється від роботи зі звичайними цілочисельними типами. Для отримання коду символу у вигляді цілого числа використовується функція `int(char)`, а для отримання символу за його кодом – `char(int)`.

➡	<pre>... // Описуємо змінні: char ch; // для символу int charCode; // для його коду</pre>
---	--

	<pre>cout << "\n Введіть код символу = "; // запрошуємо до вводу cin >> charCode; // вводимо код символу cout << "\n Код символу = " << charCode << endl; // виводимо ch = char(charCode); // отримуємо символ згідно коду cout << "\n А сам символ: " << ch << endl; // виводимо символ . . .</pre>
☞	<pre>Введіть код символу = 81 Код символу = 81 А сам символ: Q</pre>

При оголошенні рядка, тобто масива символів необхідно передбачати місце під термінатор рядка, відводячи на один символ більше.

```
char buf[10]; // масив під рядок з 9 символів + символ '\0'
```

При оголошенні символьний масив може ініціалізуватися.

```
// Оголошуємо символьну змінну і масиви:
char ch = 'A'; // ініціалізуємо символьну змінну
char b[10] = "See You 1"; // ініціалізуємо символьний масив, де звернення
// до елементів: b[0] = 'S', b[1] = 'e', . . .
char c[] = "I can see the light"; // тут термінатор сам визначає
// кількість елементів масиву й
// вставляє термінатор рядка '\0'
char d[] = {'B', 'l', 'u', 'e', '\0'}; // якщо самі формуємо рядок,
// то самі й вставляємо '\0'
```

Якщо не урахувати термінатор рядка, то це викликає несподіванки.

☞	<pre>. . . // Описуємо змінні: char ch1[] = {'a', 'b', '\0'}; // урахуємо термінатор char ch2[] = {'a', 'b'}; // не урахуємо термінатор cout << "\n ch1[] = " << ch1 << endl; cout << "\n ch2[] = " << ch2 << endl; . . .</pre>
☞	<pre>ch1[] = ab ch2[] = abab</pre>

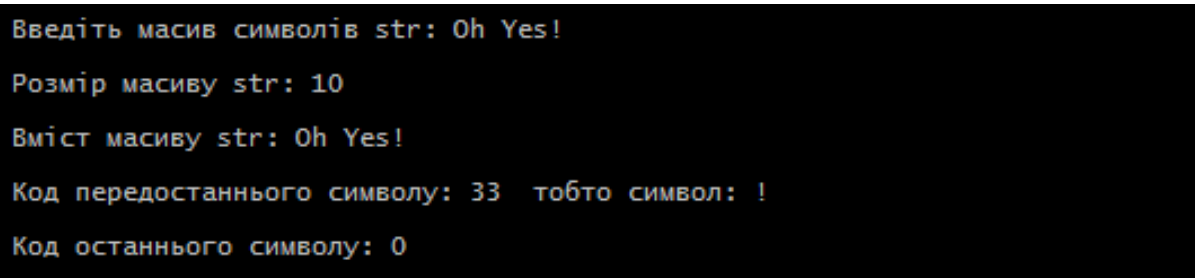
Вводити рядки за допомогою об'єкта `cin` не доцільно, оскільки він діє до першого пробілу, а наступні символи не враховує. Тому необхідно використовувати функцію.

```
cin.getline(string, size, stop)
```

де `string` – ім'я масиву символів, яку потрібно ввести;
`size` – кількість символів, які вводяться \leq довжині масива;
`stop` – символ- обмежувач, яким за умовчанням є перевід рядка (`'\n'` – Enter), інакше указуємо явно, наприклад – `'.'`, `';`, `'e'` і таке інше.

Як правило, для визначення розміру масиву символів використовується функція `sizeof()`, яка повертає розмір об'єкту в байтах.

Наприклад:

➡	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: char str[10]; // масив символів cout << "\n Введіть масив символів str: "; // запрошуємо до вводу cin.getline(str, sizeof(str)); // вводимо масив символів, // для закінчення вводу натискаємо Enter // Виводимо характеристики масиву: cout << "\n Розмір масиву str: " << sizeof(str) << endl; cout << "\n Вміст масиву str: " << str << endl; // Дивимось, що там у кінці: cout << "\n Код передостаннього символу: " << int(str[6]) << " тобто символ: " << str[6] << endl; cout << "\n Код останнього символу: " << int(str[7]) << endl; cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main </pre>
Ⓒ	 <pre> Введіть масив символів str: Oh Yes! Розмір масиву str: 10 Вміст масиву str: Oh Yes! Код передостаннього символу: 33 тобто символ: ! Код останнього символу: 0 </pre>

8.3. Завдання до роботи з символьними масивами

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Варіант
1.	З клавіатури вводяться символи доти, поки не зустрінеється крапка. Підрахувати кількість уведених символів, букв і цифр, що зустрілися у введеній фразі.
2.	З клавіатури вводиться рядок, що складається з англійських слів, набраних заголовними буквами і розділених пропусками (одним або декількома). Перетворити кожне слово в рядку, замінивши в ньому всі подальші входження його першої букви на символ «.» (крапка). Наприклад, слово «MINIMUM» треба перетворити в «MINI.U.». Кількість пропусків між словами не змінювати.
3.	З клавіатури посимвольно вводиться речення англійською. Підрахувати кількість голосних і приголосних букв у ньому та обчислити їх відсотковий вміст.
4.	З клавіатури вводиться послідовність символів доти, поки не зустрінеється символ, що не є цифрою. Підрахувати кількість уведених символів.
5.	З клавіатури вводиться послідовність символів. Якщо символ - мала літера алфавіту, то вивести на екран відповідну йому велику літеру. Уведення символів припинити, якщо зустрінеється символ '#'.
6.	З клавіатури вводиться довільне ціле число від 33 до 255. Вивести на екран символ, що відповідає цьому коду. Програма повинна припинити роботу з введенням числа 256.
7.	З клавіатури посимвольно вводиться речення. Визначити, воно розповідне, питальне чи окличне і вивести відповідне повідомлення на екран. Кінцем речення вважати знаки: '?', '!', і '.'.
8.	З клавіатури вводяться символи доти, поки речення не закінчиться. Кінцем речення вважати символи: '!', '?'. Підрахувати кількість уведених символів, відмінних від пробілу.
9.	З клавіатури вводиться фраза доти, поки не зустрінеється крапка. Підрахувати кількість слів у введеній фразі. Вважати, що слова відокремлюються одне від одного пробілами, комами, двокрапкою, крапкою з комою, тире.

№	Варіант
10.	З клавіатури посимвольно вводиться речення, в якому визначити кількість слів із закінченням на 'я' та на 'а'.
11.	З клавіатури посимвольно вводиться речення, де визначити кількість слів, які не містять букв 'и', 'т' та 'ф'.
12.	З клавіатури посимвольно вводиться фраза. Визначити, скільки в ній власних назв, тобто слів з великої літери та не власних назв. Фраза вважається введеною, якщо подальший уведений символ -крапка.
13.	З клавіатури посимвольно вводиться речення. Якщо воно починається зі слова 'Коли' або 'Тоді', вивести число 1001, а якщо з іншого слова, то вивести на екран повідомлення: 'Будь ласка, введіть інше речення'.
14.	З клавіатури посимвольно вводиться рядок, що складається з англійських слів, набраних заголовними буквами і розділених пропусками (одним або декількома). Вивести рядок, що містить ці ж слова, але розділені одним пропуском і розташовані в алфавітному порядку.
15.	У введеному рядку замінити всі коми, тире й двокрапки на крапки, а крапки – на знаки оклику. Підрахувати кількість таких заміन.
16.	У рядок через пропуск записано не більше 10 окремих слів. Записати кожне слово в окрему рядкову змінну і вивести на екран кожне слово в окремому рядку.
17.	Записати в сім рядкових змінних стільки ж прислів'їв. Підрахувати, у кількох з них зустрічається слово 'робота' або однокореневі з ним слова.
18.	У кожній з 10 рядкових змінних міститься назва художнього твору і прізвище автора. Вивести на екран наявні серед них твори Т.Г. Шевченка. Підрахувати їх кількість.
19.	У рядкову змінну через пропуск записані марки закордонних автомобілів. Замінити їх в цьому рядку <i>Mersedes</i> на " <i>Запорожець</i> ", <i>Opel</i> на " <i>Жигулі</i> ", <i>BMW</i> на " <i>Таврію</i> ", <i>Fiat</i> на " <i>Нива</i> ", <i>Shkoda</i> на " <i>Волинь</i> ".
20.	Рядок містить деяке слово. Записати в другий рядок (через пропуск) коди символів вихідного рядка, а в третій – саме слово, знаки ' ->' і коди символів вихідного рядка (наприклад, 'АВВА -> 65 66 66 65').
21.	Дано ціле додатне п'ятизначне число. Вивести символи, що зображають цифри цього числа (в порядку справа наліво).
22.	Даний рядок, що зображає ціле п'ятизначне додатне число. Вивести суму цифр цього числа.
23.	Дано рядок з чотирьох цифр, що зображує арифметичний вираз <цифра>±<цифра>±<цифра>, де на місці знака дії знаходиться символ «+» або «-» (наприклад, «4+7-2-8»). Вивести значення цього виразу (ціле число).
24.	Дано рядок, що зображує двійковий запис цілого додатного числа з чотирьох знаків. Вивести рядок, що зображує десятковий запис цього ж числа.

№	Варіант
25.	Дано рядок, що зображує десятковий запис цілого додатного числа з чотирьох знаків. Вивести рядок, що зображує двійковий запис цього ж числа.

8.4. Приклад розв'язання завдання до лабораторної роботи

	Створити програму, яка визначає, на яких розкладках клавіатури вводився текст.
--	--

Відповідний код надано нижче.

➡	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: char inputString[80]; // рядок для вводу тексту bool cyr = false, // прапор введення Кирилиці - «опущено» eng = false; // прапор введення Латиниці - «опущено» int index = 0; // змінна для перебору символів у рядку // Виводимо завдання: cout << " Створити програму, яка визначає, на яких \n"; cout << " розкладках клавіатури вводився текст. \n"; cout << "\n Введіть рядок: "; // виводимо запрошення до вводу // Вводимо рядок з клавіатури: cin.getline(inputString, sizeof(inputString)); // Перебираємо символи у рядку: while (inputString[index] != '\0') // до символа '\0' { // Перевіряємо потрапляння у Кирилицю: if ('A' <= inputString[index] && inputString[index] <= 'я') cyr = true; // піднімаємо прапор Кирилиці // Перевіряємо потрапляння Латиницю: if ('A' <= inputString[index] && inputString[index] <= 'z') eng = true; // піднімаємо прапор Латиниці ++index; // отримуємо індекс наступного символу } cout << endl; // переводим строку </pre>
---	---

	<pre>// Визначаємо розкладки клавіатури: if (cyr && eng) // якщо обидва прапори піднято, // Друкуємо обидві: cout << " Кирилиця і Латиниця." << endl; else // інакше if (cyr) // піднято флаг Кирилиця // Друкуємо Кирилиця: cout << " Кирилиця." << endl; else // інакше піднято флаг Латиниця // Друкуємо Латиниця: cout << " Латиниця." << endl; // Вивід введеного рядка: cout << endl << " Введений рядок: " << inputString << endl; return 0; // успішне закінчення програми } // кінець функції main</pre>
☐	<p>Створити програму, яка визначає, на яких розкладках клавіатури вводився текст.</p> <p>Введіть рядок: string рядок</p> <p>Кирилиця і Латиниця.</p> <p>Введений рядок: string рядок</p>
	<p>Створити програму, яка визначає, на яких розкладках клавіатури вводився текст.</p> <p>Введіть рядок: рядок</p> <p>Кирилиця.</p> <p>Введений рядок: рядок</p>
	<p>Створити програму, яка визначає, на яких розкладках клавіатури вводився текст.</p> <p>Введіть рядок: string</p> <p>Латиниця.</p> <p>Введений рядок: string</p>

Пам'ятайте. Як сказав відомий канадський вчений-інформатик Брайан Керніган, який працював у Bell Labs та сприяв розвитку операційної системи Unix разом із її творцями Кеном Томпсоном та Деннісом Річі, а його ім'я стало широко відоме завдяки співавторству у першій книзі про мову програмування C (The C Programming Language) з Деннісом Річі: «Найбільш ефективним інструментом відлагодження програми як і раніше є ретельне опрацювання коду в поєднанні з розумно розміщеними операторами друку».

Лабораторна робота №9

Тема: Основи роботи з функціями для роботи з рядками в C++

Мета: освоєння роботи з функціями для роботи з рядками. Використання функцій для обробки рядків.

Теоретична частина

9.1. Визначення функції. Переваги застосування функцій

Функції оголошені в бібліотеках `<string.h>` та `<cstring>` широко використовуються, оскільки будучи частиною стандартної бібліотеки, вони гарантовано працюють на всіх платформах, що підтримують мови C та C++. Однак, треба враховувати, що ці строкові функції працюють тільки з набором символів ASCII або його сумісними розширеннями, такими як ISO 8859-1; багатобайтові кодування такі як UTF-8 будуть працювати, з відзнакою, що «довжина» рядку буде визначатися як число байтів, а не число символів Юнікоду, яким вони відповідають. Несумісні з ASCII рядки зазвичай обробляються кодом описаним як `wchar.h`.

9.2. Функції для роботи з рядками

9.2.1. Визначення довжини рядків. Функція `strlen(S)`

Часто при роботі з рядками необхідно знати, скільки символів містить рядок. Щоб з'ясувати інформацію про довжину рядка у заголовковому файлі `<cstring>` описано функцію `strlen(S)`. Ця функція в якості єдиного параметра приймає адресу початку символьного масиву `S`, обчислює кількість символів рядка і повертає його у вигляді беззнакового цілого числа. Функція `strlen()` повертає значення на одиницю менше, ніж відводиться під масив через резервування місця для символу `'\0'`.

```
char S [ ] = "0123456789"; // оголошуємо та ініціалізуємо рядок
                        // довжиною 10 символів
cout << "Довжина = " << strlen( S ) << endl; // виводимо "Довжина = 10"
cout << "Розмір =" << sizeof( S ) << endl; // виводимо "Розмір = 11"
```

9.2.2. Копіювання рядків. Функція `strcpy(str1, str2)`

Ця функція виконує побайтне копіювання символів з рядка з адресою `str2`, до рядка з адресою `str1`, починаючи з нульового байту. Копіювання припиняється після досягнення нуль-термінатора рядка `str2`, тому перед копіюванням необхідно впевнитися, що довжина `str2` менше або дорівнює

довжині `str1`. В іншому випадку можливе виникнення помилок, пов'язаних з накладанням даних. Попередня інформація знищується!

Наприклад, наступний фрагмент копіює до рядка `S1` спочатку значення рядка «12345», а потім – «Подивимися».

```
char S1 [21]; // оголошуємо рядок довжиною 20 символів
char S2 [ ] = "12345"; // оголошуємо та ініціалізуємо
                        // рядок довжиною 5 символів
strcpy (S1 , S2); // копіюємо рядок S2 до рядка S1
cout << S1 << endl; // виводимо рядок S1 = «12345»
cout << endl; // переводимо рядок
strcpy ( S1 , "Подивимися"); // копіюємо рядок "Подивимися" в рядок S1
cout << S1 << endl; // виводимо рядок S1 = "Подивимися"
```

Можна копіювати не весь рядок, а лише окремий фрагмент (до кінця рядка `'\0'`). При цьому другий параметр є покажчиком на деякий елемент рядкового масиву. Слід мати на увазі, що ідентифікатор рядка фактично є покажчиком на початок рядка. Наприклад, наступний фрагмент скопіює до `S4` закінчення рядка `S3`.

```
char S3 [21] = "Копія рядка"; // оголошуємо та ініціалізуємо рядок
                        // довжиною 20 символів
char S4 [21]; // оголошуємо рядок довжиною 20 символів
char* pS = S3 ; // оголошуємо покажчик на рядок S3
cout << pS << endl; // виводимо рядок "Копія рядка"
pS += 6; // збільшуємо адресу pS на 6 байт
cout << pS << endl; // виводимо рядок "рядка"
strcpy ( S4 , pS ); // копіюємо рядок "рядка" в рядок S4
cout << S4 << endl; // виводимо рядок "рядка"
```

9.2.3. Копіювання частини рядка. Функція `strncpy(str1, str2, n)`

Ця функція відрізняється від `strcpy(str1, str2)` тим, що в її параметрах додається ще один аргумент `n`, який вказує кількість символів, більше якої не буде проводитись копіювання символів. У наведеному нижче фрагменті з рядка `sLong` до рядка `sShort` скопійовано чотири перші символи із затиранням вихідного значення початку короткого рядка.

```
char sLong [ ] = "0123456789"; // оголошуємо та ініціалізуємо рядок
                        // довжиною 10 символів
char sShort [ ] = "abcdef"; // оголошуємо та ініціалізуємо рядок
                        // довжиною 6 символів
strncpy ( sShort, sLong, 4); //копіюємо рядок «0123»
                        // до початку рядка sShort
cout << sShort << endl; // виводимо рядок "0123ef"
```

9.2.4. Приєднання рядка. Функція `strcat(str1, str2)`

У результаті роботи цієї функції вміст рядка, з адресою `str2`, приєднується до вмісту рядка з адресою `str1`. При цьому розмір рядкового масиву `str1` повинен бути достатнім для зберігання об'єднаного рядка.

У наступному прикладі рядок `R` ініціалізується за допомогою функції копіювання `strcpy()` і доповнюється підрядком, з використанням функції `strcat()`.

```
char R [26]; // оголошуємо рядок довжиною 25 символів
strcpy (R , "Press any key "); // ініціалізуємо рядок
strcat (R , "to continue"); // додаємо до кінця рядка "to continue"
cout << R << endl; // виводимо рядок "Press any key to continue"
```

9.2.5. Приєднання частини рядка. Функція `strncat(str1, str2, n)`

Функція копіює `n` символів рядка `str2` до кінця рядка `str1`. При цьому розмір рядкового масиву `str1` повинен бути достатнім для зберігання об'єднаного рядка. Наступний приклад виконує конкатенацію рядка `str1` з двома першими символами підрядка `str2`.

```
char R1 [31]= "Press any key "; // оголошуємо та ініціалізуємо
char R2 [31]= "to continue"; // два рядки довжиною 30 символів
strncat ( R1, R2 , 2 ); // додаємо два перші символи рядка S2
// наприкінці рядка S1
cout << R1 << endl; // виводимо "Press any key to"
```

9.2.6. Функції перетворення символів та перевірки діапазону

Заголовний файл `<cctype>` містить функції обробки символів. Ця бібліотека оголошує набір функцій для виконання різних класифікацій та деяких операцій перетворення окремих символів.

Всі функції, визначені в заголовному файлі `<cctype>`, приймають як аргумент значення типу `int` (еквівалент одного символу) і повертають ціле число, яке є еквівалентом символу, або значення, яке представляє логічне значення: цілочисельне значення 0 – «хибність», і цілочисельне значення, відмінне від 0 – «істина». У цьому файлі визначено два набори функцій:


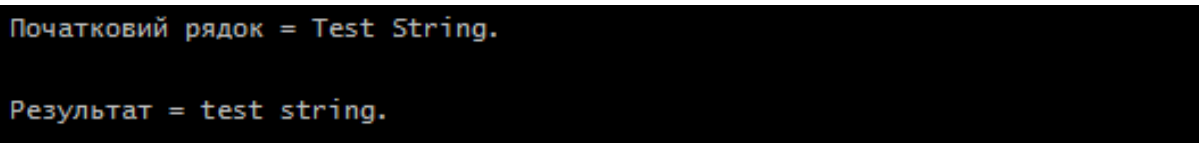
- функції перетворення символів;
- функції класифікації символів.

Перший набір функцій – це функції, що виконують деякі перетворення символів. Їх всього дві:

- `tolower()` – перетворення заголовного символу на рядковий;
- `toupper()` – перетворення рядкового символу на заголовний.

Наприклад:

```
➔ #include <iostream> // підключаємо бібліотеку вводу-виводу
#include <windows.h> // підключаємо заголовний файл локалізації
```

	<pre> #include <cctype> // підключаємо для функції tolower using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int i = 0; // індекс символів рядка char str[]="Test String.\n", // початковий рядок str1[30]; // результуючий рядок cout << "\n Початковий рядок = " << str << endl; // Організовуємо цикл по рядку: while (str[i]) // поки не кінець рядка { str1[i] = tolower(str[i]); // перетворюємо поточний // символ рядка до малих літер i++; // отримуємо індекс наступного символу } str1[i] = '\0'; // завершуємо рядок нуль-терміном cout << "\n Результат = " << str1 << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
	

До другої групи відносяться такі функції:

- `isalnum()` – функція повертає істинне значення `true`, якщо її аргумент – буква або цифра, і `false` («хибність») в інших випадках;
- `isalpha()` – функція повертає істинне значення `true`, якщо її аргумент – буква, і `false` («хибність») в інших випадках;
- `isdigit()` – функція повертає істинне значення `true`, якщо її аргумент – десяткова цифра, і `false` («хибність») в інших випадках;
- `islower()` – функція повертає істинне значення `true`, якщо її аргумент – мала літера алфавіта, і `false` («хибність») в інших випадках;
- `ispunct()` – функція повертає істинне значення `true`, якщо її аргумент – знак пунктуації, і `false` («хибність») в інших випадках;

– `isspace()` – функція повертає істинне значення `true`, якщо її аргумент – будь-який знак пробіла, і `false` («хибність») в інших випадках;

– `isupper()` – функція повертає істинне значення `true`, якщо її аргумент – велика буква алфавіта, і `false` («хибність») в інших випадках.

Наприклад:

➔	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int i = 0; // індекс символів рядка char str[]="C++"; // початковий рядок cout << "\n Початковий рядок = " << str << endl; // Організовуємо цикл по рядку: while (str[i]) // поки не кінець рядка { if (isalpha(str[i])) // якщо поточний символ рядка буква cout << " Символ " << str[i] << " - буква\n"; else // інакше cout << " Символ " << str[i] << " - не буква\n"; i++; // інкремент лічильника } // кінець while return 0; // успішне закінчення програми } // кінець функції main</pre>
⬅	<pre>Початковий рядок = C++ Символ C - буква Символ + - не буква Символ + - не буква</pre>

9.2.7. Пошук символів

9.2.7.1. Функція `strchr()`

Функція `strchr(string, symbol)` виконує пошук першого входження символу `symbol` в рядок `string`. Повертає покажчик на перше входження символу в рядку (його адресу). Завершальний нульовий символ вважається частиною рядка. Таким чином, він також може бути знайдений для отримання покажчика на кінець рядка.

➔	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу</pre>
---	---

	<pre> #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: char lotr[] = "_-=Волѡдар кілець=-_"; // рядок для пошуку символа 0 int i; // індекс шуканого символу char *ring; // адреса шуканого символу cout << "\n Шукайте кільце всевладдя в LOTR!!! \n"; ring = strchr(lotr, '0'); // знаходимо адресу символу 0 в рядку lotr cout << "\n Воно знаходиться на " << (ring - lotr + 1) << " місці \n"; // знаходимо позицію символа i = ring - lotr; // знаходимо індекс символу 0 в lotr cout << "\n Шуканий символ = " << lotr[i] << endl; // виводимо символ return 0; // успішне закінчення програми } // кінець функції main </pre>
☞	<pre> Шукайте кільце всевладдя в LOTR!!! Воно знаходиться на 7 місці Шуканий символ = 0 </pre>

9.2.7.2. Функція strstr()

Функція strstr(string1, string2) виконує пошук символів рядка string2 в рядку string1. Повертає довжину початкової ділянки рядка string1 яка складається тільки з символів, які є частиною рядка string2.

☞	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід </pre>
---	--

	<pre>// Описуємо змінні: char strtext[] = "12-й рік"; // рядок для пошуку char digit[] = "1234567890"; // рядок для порівняння int num; // кількість знайдених збігів cout << "\n Рядок для пошуку: " << strtext << endl; cout << "\n Рядок для порівняння: " << digit << endl; num = strspn(strtext, digit); // пошук рядка у підрядку cout << "\n Кількість знайдених збігів: " << num << endl; // виводимо результат return 0; // успішне закінчення програми } // кінець функції main</pre>
☞	<pre>Рядок для пошуку: 12-й рік Рядок для порівняння: 1234567890 Кількість знайдених збігів: 2</pre>

9.2.8. Пошук підрядків. Функція strstr()

Функція strstr(string1, string2) шукає перше входження підрядка string2 в рядку string1. Повертає вказівник на перше входження рядка string2 в рядок string1, або порожній вказівник, якщо рядок string2 не є частиною рядка string1. У даному пошуку нуль-символ не враховується.

☞	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: char str[] = "Ця програма написана у CodeBlocks"; // рядок для пошуку cout << "\n Рядок в якому ведемо пошук: " << str << endl; cout << "\n Друкуємо рядок з початку входження підрядка: \n"; cout << "\n " << strstr(str, "CodeBlocks") << endl; // пошук підрядка return 0; // успішне закінчення програми</pre>
---	---

	} // кінець функції main
☞	Рядок в якому ведемо пошук: Ця програма написана у CodeBlocks Друкуємо рядок з початку входження підрядка: CodeBlocks

9.2.9. Перетворення рядка на число

9.2.9.1. Функція atoi()

Функція `atoi(string)` перетворить рядок `string` в ціле значення типу `int`. Аналізуючи рядок `string`, функція інтерпретує її зміст, як ціле число, яке повертається як `int`.

Функція спочатку відкидає символи пробілів доти, поки не буде знайдений символ відмінний від нуля. Потім, починаючи з цього символу, функція приймає необов'язковий початковий знак плюс або мінус. Після чого, слід послідовність цифр, яка інтерпретується в числове значення.

Рядок може містити інші символи після ліченого цілого числа, ці символи ігноруються і ніяк не впливають на поведінку цієї функції.

Якщо перша послідовність перестав-пробільних символів у рядку `string` не є цілим числом, або, якщо `string` порожня або містить тільки пробільні символи, перетворення не виконується.

У разі успіху, функція повертає ціле число перетворене до типу `int`.

Якщо в рядку не було знайдено ціле число, функція повертає нульове значення.

Існує не стандартна ситуація, коли перетворене значення вийде з діапазону прийнятих значень типу даних `int`. Тому, передбачена більш надійна крос-платформна альтернатива – функція `strtol`, яка перетворює строку на число типу `long int`.

Наприклад:

☞	<pre>#include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: char input [81]; // рядок для перетворення int value; // отримане число</pre>
---	---

	<pre> cout << "\n Введіть ціле число: "; // запрошуємо до вводу cin >> input; // вводимо рядок value = atoi(input); // перетворюємо рядок на число cout << "\n Введене число = " << value // виводимо ціле число << "\n\n Подвоїмо його = " << (value * 2) << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
☞	<pre> Введіть ціле число: -405 Введене число = -405 Подвоїмо його = -810 </pre>

9.2.9.2. Функція atof()

Функція `atof(string)` перетворить рядок в значення типу `double`. Функція спочатку відкидає прогалини в міру необхідності, доти, поки не буде знайдений перший символ, відмінний від символу пробілу. Потім, починаючи з цього символу, `atof()` бере стільки символів, скільки можливо. Тобто, поки літерал в рядку нагадує синтаксис чисел з плаваючою точкою, функція його зчитує і інтерпретує в числове значення. Інша частина рядка, після останнього допустимого символу ігнорується і ніяк не впливає на поведінку цієї функції.

Допустиме число з плаваючою точкою формується функцією `atof()` з наступних символів:

- знаки плюс і мінус;
- послідовність чисел, які можуть містити *десяткову точку*;
- число записане в експоненційній формі, така форма може містити символи `e` або `E` за якими слідує знаки, плюс або мінус, і послідовність чисел.

Якщо перша послідовність перестав пробільних символів у рядку `string` не формують правильне число з плаваючою точкою, або рядок `string` містить тільки пробільні символи, то перетворення рядка в число не виконується.

У разі успіху, функція повертає перетворене число з плаваючою точкою типу `double`.

Якщо функції не вдалося перетворити рядок в число, повертається значення – нуль (0,0).

Наприклад:

☞	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації </pre>
---	---

	<pre> #include <cctype> // підключаємо для функції isalpha #include <cmath> // підключаємо математичну бібліотеку using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: double pi = 3.1415926535; // число ПІ char input [81]; // рядок для перетворення double agl; // само число cout << "\n Введіть кут в градусах: "; cin >> input ; agl = atof(input); // перетворюємо рядок до типу даних double cout << "\n Сінус кута " << agl << " градусів в радіанах = " // обчислюємо синус кута, переведеного в радіани << sin(agl * pi / 180) << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
☞	<pre> Введіть кут в градусах: 25.46 Сінус кута 25.46 градусів в радіанах = 0.429881 </pre>

9.2.10. Перетворення числа на рядок


9.2.10.1. Функція itoa()

Функція `itoa(value, string, radix)` перетворює ціле число `value` в рядок `string` за основою `radix`. До цифер числа `value` підбираються ANSI символи типу `char` і записуються в рядок `string`.

Функція має незручність, що полягає в тому, що при числі, що має кількість цифр, більше, ніж розмір рядка, функція видасть помилку. Тому потрібно встановлювати розмір з запасом.


Наприклад:

☞	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { </pre>
---	--

	<pre> SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int value; // десяткове ціле число char string[6] = ""; // рядок для перетвореного числа cout << "\n Введіть ціле число: "; // запрошуємо до вводу cin >> value; // вводимо ціле число itoa(value, string, 10); // перетворюємо за основою 10 cout << "\n Введене число при radix = 10: " << string << endl; itoa(value, string, 16); // перетворюємо за основою 16 cout << "\n Введене число при radix = 16: " << string << endl; return 0; // успішне закінчення програми } // кінець функції main </pre>
	<pre> Введіть ціле число: 472 Введене число при radix = 10: 472 Введене число при radix = 16: 1d8 </pre>

9.2.10.2 Функція gcvt()

Функція `gcvt(value, ndigit, string)` перетворює параметр `value` на рядок `string` довжиною `ndigit`. Наприклад:

	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: char str[25]; // double value; // int ndigit = 6; // cout << "\n Введіть дійсне число: "; // запрошуємо до вводу cin >> value; // вводимо дійсне число </pre>
---	---

	<pre>// Перетворюємо дійсне число: gcvt (value, ndigit, str); // отримуємо рядок з числом cout << "\n Отриманий рядок: " << str << endl; return 0; // успішне закінчення програми } // кінець функції main</pre>
☞	<pre>ВВедіть дійсне число: 234.5 Отриманий рядок: 234.5</pre>

9.3. Завдання до роботи з функціями роботи з рядками

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Завдання
1.	Дано ціле число $N (> 0)$ і рядок S . Перетворити рядок S на рядок довжини N так: якщо довжина рядка S більша за N , то відкинути перші символи, якщо довжина рядка S менша за N , то до її початку додати символи '.' (крапка).
2.	Дані цілі додатні числа $N1$ і $N2$ та рядки $S1$ і $S2$. Отримати з цих рядків новий рядок, що містить перші $N1$ символів рядка $S1$ і останні $N2$ символів рядка $S2$ (у зазначеному порядку).
3.	Дано символ C і рядок S . Подвоїти кожне входження символу C до рядка S .
4.	Дано символ C і рядки S , $S0$. Перед кожним входженням символу C у рядок S вставити рядок $S0$.
5.	Дані рядки S і $S0$. Перевірити, чи міститься рядок $S0$ в рядку S . Якщо міститься, то вивести True, якщо не міститься, то вивести False.
6.	У рядкову змінну пробіл записано числа. Знайти їх суму і перетворити рядок до виду: Число + Число +... + Число = Результат.
7.	Дані рядки S і $S0$. Вилучити з рядка S перший підрядок, що відповідає $S0$. Якщо немає відповідних підрядків, вивести рядок S без змін.
8.	Дані рядки S і $S0$. Вилучити з рядка S останній підрядок, що відповідає $S0$. Якщо немає відповідних підрядків, вивести рядок S без змін.

№	Завдання
9.	Дані рядки S і $S0$. Вилучити з рядка S всі підрядки, що збігаються з $S0$. Якщо немає відповідних підрядків, вивести рядок S без змін.
10.	Даний текстовий рядок, що містить слова та цілі числа, розділені пробілом. Необхідно всі цифри зменшити в 10 разів.
11.	Дано два рядки. Вилучити з кожного з них символи, відсутні в іншому.
12.	У рядкову змінну пробіл записано числа. Знайти їх твір і перетворити рядок до виду: Число * Число *... * Число = Результат.
13.	Дано рядок-речення англійською мовою. Вивести найдовше слово в реченні. Якщо таких слів декілька, то вивести перше з них. Словом вважати набір символів, який не містить пробілів, знаків перепинання і обмежений пробілами, знаками перепинання або початком/кінцем рядка.
14.	Рядок містить певне слово у нижньому регістрі. Записати до іншого рядка це слово у верхньому регістрі знак «->» та коди символів через пробіли. Наприклад, «ABBA -> 65 66 66 65»).
15.	Рядок, що містить повну назву файлу, тобто назву диска, список каталогів (шлях), власне ім'я і розширення. Вибрати з цього рядка назву файлу (без розширення).
16.	Дано рядок, що містить повну назву файлу, тобто назву диска, список каталогів (шлях), власне ім'я і розширення. Виділити з цього рядка суфікс файлу (без попередньої точки).
17.	Дано рядок, що містить повну назву файлу. Вибрати з цього рядка назву першого каталогу (без символів «\»). Якщо файл міститься в кореневій теці, вивести символ «\».
18.	Дано рядок-речення англійською мовою. Зашифрувати її, виконавши циклічну заміну кожної літери на наступну за нею в алфавіті і зберігши при цьому регістр букв («А» перейде в «В», «а» – в «b», «В» - в «С», «z» – в «a» тощо). Знаки перепинання і пробіли не змінювати.
19.	Дано рядок-речення англійською мовою і число K ($0 < K < 10$). Зашифрувати рядок, виконавши циклічну заміну кожної літери на букву того ж регістру, розташовану в алфавіті на K -й позиції після шифрованої літери (наприклад, для $K = 2$ «А» перейде в «С», «а» – в «с», «В» – в «D», «z» – в «b»). Знаки перепинання і пробіли не змінювати.
20.	Дано рядок-речення. Зашифрувати її, розмістивши спочатку всі символи, розташовані на парних позиціях рядка, а потім, всі символи, розташовані на непарних позиціях (наприклад, рядок «Something» перетвориться на «oehnSmtig»).

№	Завдання
21.	Дано рядок, що містить цифри і рядкові латинські літери. Якщо літери у рядку впорядковані за алфавітом, то вивести 0; в іншому випадку вивести номер першого символу рядка, що порушує алфавітний порядок.
22.	Дано рядок, що містить латинські літери і круглі дужки. Якщо дужки розставлені правильно (тобто кожній відкриваючій відповідає одна закриваюча), вивести число 0. В іншому випадку вивести або номер позиції, в якій розташована перша помилкова дужка закриття, або, якщо закриваючих дужок не вистачає, число -1.
23.	Дано рядок, що містить латинські літери і дужки трьох видів: «()», «[]», «{}». Якщо дужки розставлені правильно (тобто кожній відкриваючій відповідає закриваюча дужка того ж виду), вивести число 0. В іншому випадку вивести або номер позиції, в якій розташована перша помилкова дужка, або, якщо закриваючих дужок не вистачає, число -1.
24.	Дано рядок-речення з надлишковими пробілами між словами. Перетворити її так, щоб між словами був рівно один пробіл.
25.	Дано рядок-речення англійською мовою. Вивести найкоротше слово в реченні. Якщо таких слів декілька, то вивести останнє з них. Словом вважати набір символів, який не містить пробілів, знаків перепинання і обмежений пробілами, знаками перепинання або початком/кінцем рядка.

9.4. Приклад розв'язування завдання до лабораторної роботи

	У рядкову змінну через пробіл записані цілі та дійсні числа. Знайти їх суму і перетворити рядок на вигляд: Число + Число +... + Число = Результат.
--	--

Нижче наведений приклад вирішення даного завдання

➡	<pre> #include <iostream> // підключаємо бібліотеку вводу-виводу #include <windows.h> // підключаємо заголовний файл локалізації #include <cctype> // підключаємо для функції isalpha #include <string.h> // підключаємо для функцій роботи з рядками #include <ctype.h> // підключаємо для перевірки відп. симв. деякому діапазону #include <stdlib.h> // підключаємо для функцій перетворення типів using namespace std; // оголошуємо простір імен std int main() // функція main виконується першою { </pre>
---	--


```

SetConsoleOutputCP(1251); // локалізуємо вивід

// Описуємо змінні:
char S1 [81] = "12 10.555 20 5.7", // початковий рядок з
числами
        S2 [10],          // рядок для окремого числа у символьному
вигляді
        reS[81];          // результуючий рядок
char Error[20];           // рядок для помилки функції strtod
char* Err = Error;        // покажчик на рядок помилки strtod

int iSymb = 0,             // індекс символу в S1 для будь-яких
маніпуляцій
        nSpace = 0,       // кількість пробілів у вихідному рядку S1
        intSum = 0,       // кінцева сума результату для цілих чисел
        n,               // числове значення окремого цілого доданку
        nDgt,            // кількість цифр для функції gcvt
        nDgtNum;         // кількість цифр в окремому числі
double doubSum = 0,       // кінцева сума результату для дійсних
чисел
        d;               // числове значення окремого дійсного
доданку
        bool flagDot;     // прапор наявності точки в дійсному числі

// Виводимо завдання:
cout << " У рядкову змінну через пробіл записані цілі \n";
cout << " та дійсні числа. Знайти їх суму і перетворити \n";
cout << " рядок на вигляд: Число + Число +... + Число = \n";
cout << "                Результат. \n";

// Одразу друкуємо заголовок для S1:
cout << "\n Початковий рядок символів: " << endl;

// Починаємо з перевантаження символів з вихідного рядка S1
// в результативний reS і паралельно рахуємо кількість
// пробілів. Для перевірки діапазону цифр 0..9
// використовуємо функцію isdigit:
while ( isdigit(S1[iSymb]) || S1[iSymb]==' ' || S1[iSymb]=='.'
)
{ // працює, поки зустрічаються цифри, точки і пробіли
    cout << S1[iSymb];          // посимвольно роздруковуємо S1
    reS[iSymb] = S1[iSymb];     // посимвольно переносимо з S1 в
reS
    // заміна пробілів на плюси
    if (S1[iSymb]==' ')        // якщо зустрівся пробіл,
    {
        reS[iSymb]= '+';      // то замінюємо на '+'
        ++nSpace;             // рахуємо пробіли
    }
    ++iSymb;                  // просуваємо індекс в рядках S1 і
reS

```

```

    }
    reS[iSymb] = '='; // поміщаємо в кінці reS '='
    ++iSymb;          // ще раз зрушуємо індекс для '\0'
    reS[iSymb] = '\0'; // добиваємо в кінці reS '\0'
    cout << endl;      // переводимо рядок


    iSymb = 0; // встановлюємо початковий індекс для подорожі по S1
    nDgt = 0;  // встановлюємо початкову кількість цифр для функції
gcvт

    // Починаємо цикл перебору наявних у S1 чисел:
    for (int i=0; i<=nSpace; ++i) // знаємо - чисел у рядку S1 на
    {                               // одиницю більше, ніж пробілів
        nDgtNum =0; // встановлюємо індекс для чергового числа в
S1
        flagDot = false; // поки не знаємо є крапка чи ні
        // Зайнялися поточним числом - переносимо з S1 в S2:
        while (isdigit(S1[iSymb]) || S1[iSymb]=='.' ) //
перевіряємо
        { // наявність цифр і точки, а символ '+' нас зупинить
            S2[nDgtNum] = S1[iSymb]; // по цифрі переносимо в S2
            // Перевірка на наявність точки:
            if ( ispunct(S2[nDgtNum])) // якщо точка є, то
                flagDot = true; // піднімаємо прапор дійсного
числа

                ++nDgtNum; // просуваємо індекс в S2
                ++iSymb;   // просуваємо індекс в S1
        } // тут число закінчилося - далі плюс
        ++iSymb; // тоді індекс просуваємо до наступного числа
        S2[nDgtNum] = '\0'; // в S2 добиваємо «термінатор»
        // Почали працювати з числом в S2:
        if (flagDot) // якщо в числі є точка -
        {           // значить це дійсне число
            d = strtod( S2, &Err ); // перетворюємо на число
            doubSum += d; // додаємо до суми
            flagDot = false; // опускаємо прапор
            --nDgtNum; // скільки цифр без урахування точки?
            // слід запам'ятати макс. кількість цифр для функції
gcvт.

            // Якщо кількість цифр у попередньому числі менша,
            if ( nDgt < nDgtNum) // ніж у поточному,
                nDgt = nDgtNum; // то оновлюємо значення
        }
        else // інакше це ціле число
        {
            n = atoi( S2 ); // перетворюємо на машинне число в n
            intSum += n; // підсумовуємо чергове число в Sum
        }
    }
    doubSum += intSum; // додаємо суму цілих до суми дійсних
    cout << endl << "Сума чисел = " << doubSum << endl;

```

	<pre>// Перетворюємо загальну суму до рядкового вигляду gcvrt(doubSum, nDgt, S2); // і поміщаємо в S2 cout << endl << "У вигляді рядка S2 = " << S2 << endl << endl; // Допишуємо в кінець результуючого рядка reS strcat(reS, S2); // загальну суму з рядка S2 cout << " Результуючий рядок: " << endl // роздруковуємо << reS << endl; // результат return 0; // успішне закінчення програми } // кінець функції main</pre>
	<pre>У рядкову змінну через пробіл записані цілі та дійсні числа. Знайти їх суму і перетворити рядок на вигляд: Число + Число +... + Число = Результат. Початковий рядок символів: 12 10.555 20 5.7 Сума чисел = 48.255 У вигляді рядка S2 = 48.255 Результуючий рядок: 12+10.555+20+5.7=48.255</pre>

Пам'ятайте. Фінсько-американський інженер-програміст, який є творцем і, історично, основним розробником ядра Linux, сказав: «Погані програмісти турбуються про код. Хороші програмісти турбуються про структури даних та їх відносини».

Лабораторна робота №10

Тема: Основи роботи з текстовими файлами в C++

Мета: освоєння роботи з текстовими файлами в C++.

Теоретична частина

10.1. Загальні поняття про файл

У програмах на C++ існує можливість роботи з файлами, які є базовими вбудованими типами об'єктів. Файл – це іменована область зовнішньої пам'яті. Керуванням роботою файлу займається операційна система. Для кожного файлу створюється об'єкт, який забезпечує зв'язок з файлом. Файл може бути розміщений в будь-якому місці комп'ютера (жорсткий диск, флеш-носій тощо).

10.2. Особливості роботи з текстовими файлами в C++

Текстові файли ще мають назву *файли послідовного доступу*, бо як запис так і читання в них проходять *послідовно*.

Файл – це іменований набір байтів, який можна зберегти на певному накопичувачі. Ну, тепер ясно, що під файлом розуміється деяка послідовність байтів, яка має своє, унікальне ім'я, наприклад `file.txt`. В одному каталозі не можуть знаходитися файли з *однаковими іменами*. Назва файлу припускає не лише одну назву файла, але й *суфікс*, наприклад: `file.txt` і `file.dat` – різні файли, хоч і мають однакові назви. Але суфікси у них різні! Існує таке поняття, як *повна назва файлів* – це *шлях* до директорії файла з вказанням назви файла, наприклад: `D:\docs\file.txt`. Важливо розуміти ці базові поняття, інакше складно буде працювати з файлами.

Для роботи з файлами необхідно підключити заголовний файл `<fstream>`. В `<fstream>` визначено декілька класів і з'єднано заголовні файли `<ifstream>` – *файловий ввід* і `<ofstream>` – *файловий вивід*.

Файловий ввід/вивід подібний до стандартного вводу/виводу (екран/клавіатура), єдина відмінність – це те, що ввід/вивід виконується з використанням файлів. Якщо ввід/вивід на стандартні пристрої виконується за допомогою об'єктів `cin` і `cout`, то для організації файлового вводу/виводу достатньо створити власні об'єкти, які можна використовувати аналогічно цим операторам.

Наприклад, необхідно створити текстовий файл і записати до нього рядок "Робота з файлами в C++". Для цього необхідно виконати наступні кроки:

- створити об'єкт класу `ofstream`;
- зв'язати об'єкт класу з файлом, в який буде проводитися запис;

- записати рядок до файла;
- закрити файл.

Щоб прочитати файл, необхідно виконати ті самі кроки, що й під час запису до файлу з невеликими змінами:

- створити об'єкт класу `ifstream`;
- зв'язати його з файлом, з якого буде проводитися зчитування;
- прочитати рядок з файлу;
- закрити файл.

Наприклад:

```
➔ #include <iostream> // підключаємо бібліотеку вводу-виводу
#include <windows.h> // підключаємо заголовний файл локалізації
// підключаємо бібліотеку читання-запису файлів:
#include <fstream> // для роботи з файлами
using namespace std; // оголошуємо простір імен std

int main() // функція main виконується першою
{
    SetConsoleCP(1251); // локалізуємо ввід
    SetConsoleOutputCP(1251); // локалізуємо вивід


    // Описуємо змінні:
    char s[20]; // символьний рядок для виводу з файла
    double num = 2.564; // число для запису в файл

    // Створюємо об'єкт класу ofstream F1:
    ofstream F1; // для запису до файла
    // Відкриваємо файл table.txt для запису:
    F1.open("G:\\Lab_9\\table.txt"); // в теку Lab_9 диску G:\
    // Записуємо в файл рядок тексту:
    F1 << "Робота з файлами в C++" << endl; // запис в файл
    // Допишуємо в файл число num:
    F1 << num << endl;
    // Закриваємо файл:
    F1.close();

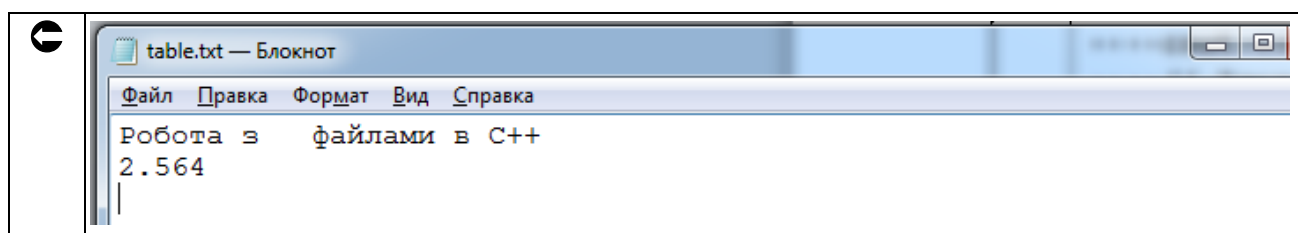
    // Створюємо об'єкт класу ifstream F2:
    ifstream F2; // для читання файла
    // Відкриваємо файл table.txt для читання:
    F2.open("G:\\Lab_9\\table.txt"); // з теки Lab_9 диску G:\

    // Виводимо заголовок:
    cout << "\n Інформація з файлу: \n" << endl;

    // Організуємо цикл для читання результатів:
    for(int i=1; i<=5; ++i) // читаємо 5 слів!!!
    {
        F2 >> s; // читаємо по слову з файлу в рядковий масив
        cout << ' ' << s ; // виводимо рядок на екран
    }
}
```

	<pre> } // Окремо читаємо з файлу: F2 >> num; // записане в файл число cout << "\n\n num = " << num; // виводимо його // Закриваємо файл: F2.close(); cout << endl; // переводимо рядок return 0; // успішне закінчення програми } // кінець функції main </pre>
	<pre> Інформація з файлу: Робота з файлами в C++ num = 2.564 </pre>

Результатом роботи програми є файл `table.txt` у теці **Lab_9** диску **G:**. Його вміст показано нижче.



Зверніть увагу! Одиницею інформації при запису чи читанні є одиниці тексту (слово) або числа розділені одним або декількома пробілами. Спеціально перед словом «файлами» стоїть 3 пробіла. Також розділювачем може бути знак переведення рядка `'\n'`.

Оскільки файл – це послідовність байтів, то *текст* в ньому зберігається як послідовність ASCII-кодів символів. Для чисел проводиться *конвертація з двійкового представлення* в оперативній пам'яті в послідовність ASCII-кодів цифр чисел і тоді слідує запис у файл. При читанні з файлу йде зворотній процес – ASCII-коди цифр кожного числа перетворюються в *двійкове представлення* чисел в оперативній пам'яті!

10.3. Форматування інформації при введенні в текстовий файл

Для форматування інформації при введенні в текстовий файл використовується заголовний файл `<iomanip>`, що містить ряд корисних маніпуляторів:

1. `setiosflags()` – задає вказані прапорці, наприклад: `setiosflags(ios::fixed|ios::showpoint)` – числа представляються у форматі з фіксованою точкою.

2. `setprecision(n)` – виводяться n знаків після коми.

Ці маніпулятори знадобляться нам для виконання завдань.

10.4. Завдання до роботи з текстовими файлами

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою Python з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Завдання
1.	У текстовому файлі – дані про довжини сторін 20 прямокутних паралелепіпедів. Обчислити об’єми цих паралелепіпедів. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину першої, другої та третьої сторін паралелепіпеда, об’єм паралелепіпеда. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $V = a \cdot b \cdot c$.
2.	У текстовому файлі – дані про довжини сторін 15 прямокутників. Обчислити периметри і площі цих прямокутників. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину першої та другої сторін прямокутника, периметр та площу прямокутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
3.	У текстовому файлі – дані про радіуси 12 кіл. Обчислити довжину кіл і площу кругів, обмежених цими колами. Результати розрахунку записати у файл як таблицю, що містить стовпці: радіус і довжину кола, площу круга. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $l = 2 \cdot \pi \cdot R$, $S = \pi \cdot R^2$.
4.	У текстовому файлі – дані про радіуси основ і висоти 12 циліндрів. Обчислити об’єм і площу бічної поверхні цих циліндрів. Результати розрахунку записати у файл як таблицю, що містить стовпці: радіус основи, висоту, об’єм та площу бічної поверхні циліндра. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $V = \pi \cdot R^2 \cdot h$, $S_{\text{бічн}} = 2 \cdot \pi \cdot R \cdot h$.

№	Завдання
5.	<p>У текстовому файлі – дані про радіуси основ і висоти 15 конусів. Обчислити об'єм і площу бічної поверхні цих конусів. Результати розрахунку записати у файл як таблицю, що містить стовпці: радіус основи, висоту, об'єм та площу бічної поверхні конуса. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $V = \frac{1}{3}\pi \cdot R^2 \cdot h$, $S_{\text{бічн}} = \pi \cdot R \cdot l$.</p>
6.	<p>У текстовому файлі – дані про катети 18 прямокутних трикутників. Обчислити гіпотенузи і площі цих трикутників. Результати розрахунку записати у файл як таблицю, що містить стовпці: перший та другий катети трикутника, а також гіпотенузу та площу трикутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $S = \frac{1}{2}a \cdot b$, $c = \sqrt{a^2 + b^2}$.</p>
7.	<p>У текстовому файлі - дані про довжини трьох сторін 14 трикутників. Обчислити площі і периметри цих трикутників. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину, другої та третьої сторін трикутника, площу та периметр трикутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.</p>
8.	<p>У текстовому файлі - дані про довжини основ і висоти 19 трапецій. Обчислити площі цих трапецій. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину нижньої та верхньої основи трапеції, висоту та площу трапеції. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.</p>
9.	<p>У текстовому файлі - дані про довжини сторін 15 рівносторонніх трикутників. Обчислити площі і периметри цих трикутників. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину сторони, периметр та площу трикутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.</p>

№	Завдання
10.	<p>У текстовому файлі - дані про кількість сторін, довжини сторін і радіуси вписаних кіл для 14 правильних багатокутників. Обчислити площі і периметри цих багатокутників. Результати розрахунку записати у файл як таблицю, що містить стовпці: кількість сторін, довжину сторони, периметр та площу багатокутника, а також радіус вписаного у багатокутник кола. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $S = \frac{n}{2} a \cdot r$.</p>
11.	<p>У текстовому файлі - дані про довжини сторін і кути між ними для 18 паралелограмів. Обчислити площі і периметри цих паралелограмів. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину першої та другої сторін паралелограма, кут між сторонами, периметр та площу паралелограма. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.</p>
12.	<p>У текстовому файлі - дані про довжини сторін і кутів між ними для 18 паралелограмів. Вибрати паралелограми, що є прямокутниками, і записати у файл таблицю, що містить таку інформацію: довжину першої та другої сторін прямокутника, периметр та площу прямокутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.</p>
13.	<p>У текстовому файлі - дані про кількість сторін, довжини сторін і радіуси вписаних кіл для 14 правильних багатокутників. Вибрати багатокутники з парною кількістю сторін і записати у файл таблицю, що містить таку інформацію: кількість сторін багатокутника, довжину його сторони, радіус вписаного в нього кола, периметр та площу багатокутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $S = \frac{n}{2} a \cdot r$.</p>
14.	<p>У текстовому файлі - дані про довжини сторін 15 рівносторонніх трикутників. Вибрати трикутники зі стороною, більшою одиниці і меншою трьох, і записати у файл таблицю, що містить таку інформацію: довжину сторони трикутника, периметр та площу трикутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.</p>

№	Завдання
15.	У текстовому файлі – дані про довжини основ і висоти 19 трапецій. Вибрати трапеції з висотою, меншою верхньої основи, і записати у файл таблицю, що містить таку інформацію: довжину нижньої та верхньої основ трапеції, висоту та площу трапеції. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
16.	У текстовому файлі – дані про довжини трьох сторін 14 трикутників. Вибрати рівносторонні трикутники і записати у файл таблицю, що містить таку інформацію: довжину сторони трикутника, площу та периметр трикутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
17.	У текстовому файлі – дані про катети 18 прямокутних трикутників. Вибрати рівнобічні трикутники і записати у файл таблицю, що містить таку інформацію: катет, гіпотенузу, та площу трикутника. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
18.	У текстовому файлі – дані про радіуси основ і висоти 15 конусів. Вибрати конуси, радіус основи яких більше висоти, і записати у файл таблицю, що містить таку інформацію: радіус основи, висоту, об'єм конуса та площу його бічної поверхні. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $V = \frac{1}{3} \pi \cdot R^2 \cdot h$, $S_{\text{бічн}} = p \cdot R \cdot l$.
19.	У текстовому файлі – дані про радіуси основ і висоти 12 циліндрів. Вибрати циліндри, радіус основи яких дорівнює висоті, та записати у файл таблицю, що містить таку інформацію: радіус основи, висоту, об'єм циліндра та площу його бічної поверхні. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $V = \pi \cdot R^2 \cdot h$, $S_{\text{бічн}} = 2 \cdot \pi \cdot R \cdot h$.
20.	У текстовому файлі – дані про радіуси 12 кіл. Вибрати кола, довжина яких більше семи, але менше 10, і записати у файл таблицю, що містить таку інформацію: радіус і довжину кола та площу круга. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора. Указівка: $l = 2 \cdot \pi \cdot R$, $S = \pi \cdot R^2$.

№	Завдання
21.	У текстовому файлі - дані про довжини сторін 15 прямокутників. Вибрати прямокутники, що є квадратами, і записати у файл таблицю, що містить таку інформацію: довжину сторони, периметр та площу квадрата. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
22.	У текстовому файлі - дані про довжини сторін 20 прямокутних паралелепіпедів. Вибрати паралелепіпеди, що є кубами, і записати у файл таблицю, що містить таку інформацію: довжину сторони та об'єм куба. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
23.	У текстовому файлі - дані про діагоналі 15 ромбів. Обчислити площі і сторони цих ромбів. Результати розрахунку записати у файл як таблицю, що містить стовпці: довжину більшої діагоналі, довжину меншої діагоналі, сторону та площу ромба. Записати відповідні заголовки для стовпців. Файл із вихідними даними сформувати за допомогою текстового редактора.
24.	У текстовий файл записані матриці $A(4 \times 4)$, $B(3 \times 3)$ і $C(5 \times 5)$. Додати до діагональних елементів кожної з них значення мінімального за модулем ненульового елемента масиву. Отримані в такий спосіб матриці вивести в інший текстовий файл. Для введення, виведення і додавання до діагональних елементів числа застосувати функції, а також використати функцію при визначенні зазначеного мінімального елемента.
25.	У текстовий файл записані матриці $A(5 \times 3)$, $B(2 \times 6)$, $C(4 \times 4)$ і $D(3 \times 5)$. Дописати в цей файл кількість додатних і від'ємних елементів масивів. Для виведення масивів та обчислення кількості зазначених елементів застосувати функції.

10.5. Приклад розв'язання завдання до лабораторної роботи

	На диску G: у теці Lab_9 у текстовому редакторі створено файл <code>radius.txt</code> , який містить чотири значення радіусу кола 1.5, 2.5, 3.6 та 4.2 в текстовому вигляді. Створити програму для обчислення довжини кола і площі кола для кожного з заданих значень. Результати розрахунків записати на диск G: у файл <code>table.txt</code> як таблицю, яка містить колонки: номер по порядку, радіус кола, довжина кола, площа кола. Попередньо до файла записати заголовки для стовпчиків.
--	--

Перед запуском програми на диску G: у теці Lab_9 у текстовому редакторі створіть файл radius.txt. Вміст текстового файлу radius.txt наведено на рисунку 10.1.

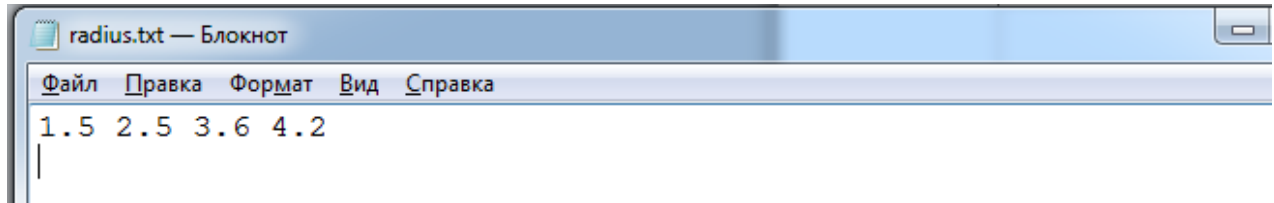


Рис. 10.1. Вміст текстового файлу radius.txt

Відповідаючи рішення даної задачі програма виглядає наступним чином.

```
➔ // Підключаємо:
#include <iostream> // бібліотеку вводу-виводу
#include <windows.h> // заголовний файл локалізації
#include <fstream> // бібліотеку читання-запису файлів
#include <iomanip> // бібліотеку для форматування потоків
using namespace std; // оголошуємо простір імен std

int main() // функція main виконується першою
{
    SetConsoleCP(1251); // локалізуємо ввід
    SetConsoleOutputCP(1251); // локалізуємо вивід

    // Описуємо змінні:
    const double pi = 3.14158; // константа пи
    double r, // радіус окружности
           l, // длина окружности
           S; // площадь круга

    // Описуємо змінну F2 для читання з файла:
    ifstream F2;
    // Відкриваємо файл radius.txt для читання:
    F2.open("G:\\Lab_9\\radius.txt");

    // Описуємо змінну F1 для запису до файла:
    ofstream F1;
    // Відкриваємо файл table.txt для запису
    F1.open("G:\\Lab_9\\table.txt");

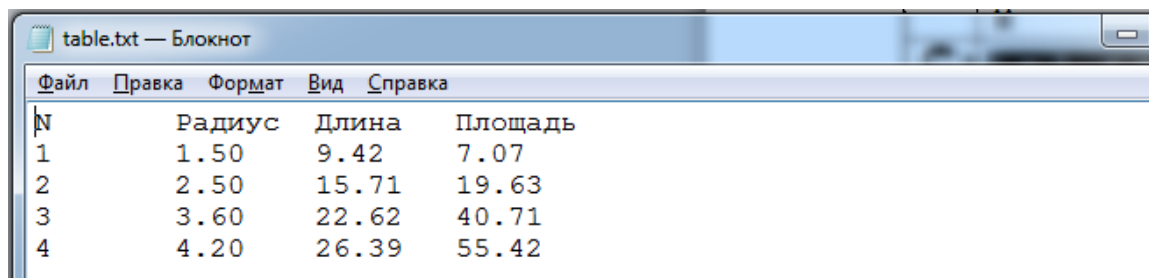
    // Вводимо у вихідний файл заголовок стовпців:
    F1 << "N \tРадіус\tДлина\tПлощадь" << endl;

    // Організуємо цикл запису результатів у вихідний файл:
    for(int i=1; i<=4; ++i)
    {
        F2 >> r; // читаємо з вхідного файла значення r
        // Роздруковуємо в циклі i та r:
        cout << "i = " << i << "   r = " << r << endl;
```

```
l = 2*pi*r; // для r обчислюємо довжину кола,  
S = pi*r*r; // а також площу круга  
// Під час запису до файла форматуємо потік,  
// причому формат з фіксованою точкою:  
F1 << setiosflags(ios::fixed|ios::showpoint)  
    << setprecision(2) // виводиться 2 знаки після коми  
    << i << '\t' // записуємо номер за порядком з табуляцією  
    << r << '\t' // записуємо радіус кола з табуляцією  
    << l << '\t' // записуємо довжину кола з табуляцією  
    << S << endl; // записуємо площу кола  
}  
  
F1.close(); // закриваємо файл для запису  
F2.close(); // закриває файл для читання  
  
// Після завершення програми у текстовому редакторі  
// дивимось вміст файла для запису table.txt.  
  
return 0; // успішне закінчення програми  
} // кінець функції main
```

```
i = 1  r = 1.5  
i = 2  r = 2.5  
i = 3  r = 3.6  
i = 4  r = 4.2
```

Перевірте вміст вихідного файлу `table.txt`, який повинен виглядати згідно рисунку 10.2.



№	Радіус	Длина	Площадь
1	1.50	9.42	7.07
2	2.50	15.71	19.63
3	3.60	22.62	40.71
4	4.20	26.39	55.42

Рис. 10.2. Вміст вихідного файлу `table.txt`

Пам'ятайте! Відомий канадський вчений-інформатик Брайан Керніган сказав: «Незважаючи ні на що, спосіб навчитися програмувати – написати код і переписати його, побачити використаний і переписати знову. Читання чужого коду також безцінне».

Лабораторна робота №11

Тема: Основи роботи зі структурами в C++

Мета: освоєння роботи зі структурами та їх використання в C++.

Теоретична частина

11.1. Визначення структур

У програмуванні є багато випадків, коли може знадобитися більше однієї змінної для представлення певного об'єкта. Наприклад, для уявлення самого себе, ви, швидше за все, захочете вказати своє ім'я, день народження, зріст, вагу або будь-яку іншу інформацію... На щастя, мова C++ дозволяє програмістам створювати власні, так звані користувацькі, типи даних – типи, які групують кілька окремих змінних разом – *структури*.

Структури C++ застосовуються для логічного об'єднання пов'язаних між собою різнорідних даних. У структуру, на протилежність масиву, можна об'єднувати дані різних типів. Оголошення структури слід розглядати як *оголошення типу*.

11.2. Опис структур

Для оголошення структури використовується ключове слово `struct`. Загальна форма оголошення структури:

```
struct ім'яСтруктури
{
    тип1 елемент1;
    тип2 елемент2;
    ...
    типN елементN;
};
```

де *ім'яСтруктури* – назва шаблону структури, який утворює новий тип даних, котрий може бути використаний у подальшому в програмі;

тип1, тип2, ... , типN – вбудовані або складені типи даних C++, які використовуються для оголошення типів елементів структури;

елемент1, елемент2, ... , елементN – імена елементів (змінних), що входять у структуру, які ще називаються *полями структури*.

Ще раз підкреслимо, що типи елементів (тобто *полів* структури) можуть бути або *вбудованими* (`int`, `double`, `char` і таке інше) або відноситися до *агрегатних* типів даних (масивів, структур, об'єднань (`union`), зчислень (`enumeration`) і класів).

Саме оголошення структури *не створює ніяких змінних*, а також при оголошенні структури *не відбувається виділення фізичної пам'яті*, як при

оголошенні змінних або масивів. Тобто, визначення структури тільки задає внутрішню організацію змінних у структурах до того, як вони будуть визначені.

Наприклад:

```
// Шаблон структури:
struct Event // розклад роботи конференц-зала
{
    int hour,          // час роботи (години)
    min;              // хвилини роботи
    char theme[100], // тема конференції
    name[100];        // ім'я та прізвище організатора
    int num;           // кількість учасників
};
```

Таким чином у нас оголошена структура з назвою Event і можна описувати змінні цього типу точно також як змінні вбудованих типів, наприклад:

```
Event conf, // структура типу Event
confAll [10]; // масив структур
```

Для звернення до поля використовується *операція вибору* – «точка».

```
// Присвоювання значень полям структури conf:
conf.hour = 12; // в змінній hour значення 12
conf.min = 30;  // в змінній min значення 30

// Присвоювання значень полям структури confAll[10]:
confAll[0].hour = 2; // в змінній hour значення 2
confAll[5].min = 30; // в змінній min значення 30
```

Структури, як і масиви, можна ініціалізувати списком значень.

```
Event conf = {12, 30, "Вирощування кактусів", 25};
```

Ввод/вивід структур, як і масивів, виконується поелементно. Наприклад, для структури conf.

```
cin >> conf.hour >> conf.min; // ввід полів hour та min
cin.getline(conf.theme, 99);   // ввід поля theme

// Вивід полів структури:
cout << "Час роботи конференції: " << conf.hour
    << "Хвилини роботи конференції: " << conf.min
    << "Тема конференції: " << conf.theme << endl;
```

11.3. Передача структур в функції

Існує 2 способи передачі структури у функцію:

– передача структури за значенням. При такій передачі робиться копія структурної змінної в пам'яті та її адреса через стек передається у функцію. Якщо структура має великий розмір, то такий спосіб неефективний. Перевагою

цього способу є те, що всі маніпуляції з копією структури у функції не впливають на вихідну змінну;

– передача покажчика на структуру. У цьому випадку через стек у функцію передається тільки покажчик на структуру, тобто її адреса в пам'яті. Якщо структура займає великий об'єм пам'яті, то такий спосіб забезпечує швидку передачу значень структурної змінної у функцію. Це пов'язано з тим, що не робиться в пам'яті копії структурної змінної.

Розглянемо приклад з передачею покажчика на структуру в функцію.

➔	<pre>// Підключаємо: #include <iostream> // бібліотеку вводу-виводу #include <windows.h> // заголовний файл локалізації using namespace std; // оголошуємо простір імен std // Описуємо шаблон структури: struct myPoint { int x; int y; }; myPoint point; // // Прототип функцій з передачею параметрів: void transByPointer(myPoint *point); // за покажчиком int main() // функція main виконується першою { SetConsoleOutputCP(1251); // локалізуємо вивід cout << "\n Передаємо параметр за за покажчиком:" << endl; transByPointer(&point); cout << " transByPointer: point.x = " << point.x << endl; return 0; // успішне закінчення програми } // кінець функції main // Функція приймає параметр за покажчиком: void transByPointer (// опис формальних параметрів: myPoint *point // значення адреси параметра){ // Зверніть увагу на роботу з полем структури! point->x = 300; // присвоюємо 300 }</pre>
☐	<pre>Передаємо параметр за за покажчиком: transByPointer: point.x = 300</pre>

При передаванні масива структур у функцію основні дії такі.

```
. . .
struct inv          // шаблон структури
{                  // для інвентаризації
    char item[30]; // назва елемента
    float cost;    // вартість елемента
    int on_hand;   // кількість наявних елементів
};

// Глобальні змінні:
int size = 4;      // розмір масиву структур
inv invInfo[4];   // масив структур

// Прототип функцій:
void enter(inv *invInfo); // введення елемента структури
. . .
int main() // главная функция
{
    . . .
    enter(invInfo); //ввод структури
    . . .
    return 0;
}
// функція вводу елемента структури
void enter(inv *invInfo)
{
    . . .
    cout << "Введіть назву: "; // запрошуємо на ввід
    cin >> invInfo[index].item; // зчитуємо ім'я
    . . .
}
```

11.4. Завдання до роботи зі структурами

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

- номер, тему і мету лабораторної роботи;
- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

Вказівка. Кожний студент обирає для завдання тільки перші 3 пункти з розділу «Інформація включає такі дані»!

№	Варіант
1.	<p>Написати програму для обробки інформації про товари, що зберігаються на складі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Найменування товару. ➤ Вартість товару. ➤ Країна-виробник товару. ➤ Кінцевий термін реалізації товару. ➤ Кількість, наявного товару. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список товарів, з закінчуючим терміном реалізації (менше місяця).
2.	<p>Написати програму для обробки інформації про жорсткі диски. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Ємність накопичувача. ➤ Тип жорсткого диска. ➤ Форм-фактор. ➤ Інтерфейс. ➤ Швидкість обертання шпинделя. ➤ Ціна. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список: компанія-виробник, тип жорсткого диска, форм-фактор (для граничних: ціни та ємності накопичувача, зазначених користувачем).
3.	<p>Написати програму для обробки інформації про країни світу. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва країни. ➤ Площа країни. ➤ Населення країни. ➤ Державна мова. ➤ Грошова одиниця. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список країн, щільність населення в яких перевищує щільність, зазначену користувачем.

№	Варіант
4.	<p>Написати програму для обробки інформації про продукти харчування. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва продукту. ➤ Калорійність продукту. ➤ Вміст у продукті вітамінів (А, В, С, Е, Р, РР). ➤ Вартість 1 кг продукту. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список продуктів, що містять вітаміни, зазначені користувачем.
5.	<p>Написати програму для обробки інформації про потяги. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Номер поїзда. ➤ Станція відправлення. ➤ Станція призначення. ➤ Час відправлення. ➤ Час прибуття. ➤ Тип поїзда (пасажирський, швидкий). <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список поїздів, що прямують до станції, зазначеної користувачем і час їх прибуття.
6.	<p>Написати програму для обробки інформації про книги. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Автор. ➤ Назва. ➤ Видавництво. ➤ Рік видання. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список книг, необхідного автора, виданих до зазначеного користувачем року.
7.	<p>Написати програму для обробки інформації про пасажирів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ ПІБ пасажирів. ➤ Номер рейсу. ➤ Реєстраційний номер багажу. ➤ Вага багажу. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список пасажирів, багаж яких перевищує приборкану користувачем вагу.

№	Варіант
8.	<p>Написати програму для обробки інформації про студентів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ ПІБ студента. ➤ Група. ➤ Стать. ➤ Екзаменаційні оцінки з вищої математики та фізики. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список студентів заданої статі, у яких оцінка з вищої математики нижче оцінки з фізики.
9.	<p>Написати програму для обробки інформації про співробітників фірми. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ ПІБ співробітника. ➤ Рік народження. ➤ Пол. ➤ Посада. ➤ Термін роботи у фірмі. ➤ Заробітна плата. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список співробітників зазначеної статі, для яких термін роботи у фірмі менше зазначеної.
10.	<p>Написати програму для обробки інформації про автомобілі, що здаються на прокат. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Марка автомобіля. ➤ Країна-виробник. ➤ Рік випуску. ➤ Довжина пробігу. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список автомобілів необхідного року випуску з необхідною довжиною пробігу.
11.	<p>Написати програму для обробки інформації про смартфони. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва смартфона. ➤ Фірма-виробник. ➤ Діагональ екрану. ➤ Оперативна пам'ять. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список смартфонів з діагоналлю екрану вище середньої.

№	Варіант
12.	<p>Написати програму для обробки інформації про твори мистецтва, виставлені на аукціонний продаж. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Номер лота. ➤ Назва твору. ➤ Автор. ➤ Країна. ➤ Початкова вартість. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список лотів, вартість яких не більше вартості, зазначеної користувачем.
13.	<p>Написати програму для обробки інформації про каталог всіх книг, видатних на абонементі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Автор (и). ➤ Назва. ➤ Обсяг (у сторінках). ➤ Кількість примірників бібліотеки. ➤ "Жанр" (навчальні посібники, класична художня література, сучасний детектив, сучасна фантастика). <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список всіх навчальних посібників, обсяг яких не більше обсягу, зазначеного користувачем.
14.	<p>Написати програму для обробки інформації про тварин, представлених на виставку котів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Кличка. ➤ Колір. ➤ Порода. ➤ Вік. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список тварин, зазначеної породи і віку.
15.	<p>Написати програму для обробки інформації про проведення вітрильної регати. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва яхти. ➤ Клас яхти (Фінн, Зоряний, Торнадо, Солінг). ➤ Порт приписки. ➤ Зайняте місце на регаті. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран інформацію про яхти відповідного класу відповідно до зазначеного користувача зайнятого місця.

№	Варіант
16.	<p>Написати програму для обробки інформації про замовлення, що надходять до годинної майстерні. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Номер замовлення. ➤ Прізвище замовника. ➤ Марка годин. ➤ Термін виконання. ➤ Вартість замовлення. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран номер замовлення і його вартість для зазначеного користувачем терміну виконання.
17.	<p>Написати програму для обробки інформації про ноутбуки. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Фірма-виробник. ➤ Тактова частота процесора. ➤ Діагональ екрану. ➤ Вартість. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран тактову частоту процесора і діагональ екрану ноутбуків, вартість яких не перевищує ціну, зазначену користувачем.
18.	<p>Написати програму для обробки інформації про пропоновані до продажу автомобілі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Країна-виробник. ➤ Марка автомобіля. ➤ Колір корпусу. ➤ Ціна автомобіля. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список автомобілів, вказаного користувачем кольору і граничної ціни.
19.	<p>Написати програму для обробки інформації про планшети. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Розмір планшета. ➤ Вага планшета. ➤ Ціна планшета. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список планшетів не великих ваги і ціни, зазначених користувачем.

№	Варіант
20.	<p>Написати програму для обробки інформації про фотоапарати. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Зум. ➤ Матриця (у пікселях). ➤ Ціна. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список даних: компанія-виробник, зум, матриця (у пікселях) для фотоапаратів, ціна яких не більша за ціну, зазначену користувачем.
21.	<p>Написати програму для обробки інформації про комп'ютерні маніпулятори типу «миша». Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Тип миші. ➤ Роз "єм для підключення. ➤ Чутливість. ➤ Кількість кнопок. ➤ Ціна. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран кількість кнопок і тип миші для граничної ціни, вказаної користувачем.
22.	<p>Написати програму для обробки інформації про відеореєстратори. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Максимальна роздільна здатність відео. ➤ Вбудований GPS. ➤ Роздільна здатність сенсора. ➤ Ціна. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран компанію-виробника і дозвіл сенсора для граничного максимального дозволу відео і ціни, зазначених користувачем.

№	Варіант
23.	<p>Написати програму для обробки інформації про відеокарти. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Графічний чіп. ➤ Частота пам'яті. ➤ Обсяг пам'яті. ➤ Тип пам'яті. ➤ Максимально підтримувана роздільна здатність. ➤ Ціна. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список: графічний чіп, частота пам'яті, обсяг пам'яті (для граничних ціни і максимально підтримуваного дозволу, зазначених користувачем).
24.	<p>Написати програму для обробки інформації про телевізори. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Діагональ екрану. ➤ Підтримка Smart TV. ➤ Дозвіл. ➤ Wi-Fi. ➤ Ціна. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список: компанія-виробник, підтримка Smart TV, дозвіл екрану (для граничних ціни і діагоналі екрану, зазначених користувачем).
25.	<p>Написати програму для ведення обліку торгових операцій фірми. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Реєстраційний номер операції. ➤ Сутність угоди (товар). ➤ Обсяг угоди (грн.). ➤ Форма розрахунку (нал., безнал., бартер). ➤ Дата вчинення. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none"> ➤ Зчитувати дані з клавіатури і заносити їх у словник. ➤ Зчитувати дані зі словника і виводити на екран список угод, обсяг яких не більше обсягу, зазначеного користувачем, а також відповідає даті здійснення, інакше видати діагностику.

11.5. Приклад розв'язання завдання до лабораторної роботи

	<p>Написати програму для обробки інформації з інвентаризації товарів, що зберігаються на складі. Інформація включає такі дані:</p> <ul style="list-style-type: none">➤ назва елемента;➤ вартість елемента;➤ кількість наявних елементів. <p>Програма повинна дозволяти користувачеві:</p> <ul style="list-style-type: none">➤ вводити елементи масиву структур;➤ виводити елементи масиву структур;➤ очищати елементи масиву структур.
--	--

Нижче наведено приклад розв'язання завдання до лабораторної роботи.

➔	<pre>// Підключаємо: #include <iostream> // бібліотеку вводу-виводу #include <windows.h> // заголовний файл локалізації #include <string.h> // бібліотеку з ф-ціями роботи з рядками using namespace std; // оголошуємо простір імен std struct inv // шаблон структури { // для інвентаризації char item[30]; // назва елемента float cost; // вартість елемента int on_hand; // кількість наявних елементів }; // Глобальні змінні: int size = 4; // розмір масиву структур inv invInfo[4]; // масив структур // Прототипи функцій: int menu_select(); // зчитування пункту меню у змінну void enter(inv *invInfo); // введення елемента структури void list(inv *invInfo); // вивід структур void del(inv *invInfo); // очищення структури int main() // функція main виконується першою { SetConsoleCP(1251); // локалізуємо ввід SetConsoleOutputCP(1251); // локалізуємо вивід // Описуємо змінні: int choice; // для вибору дії bool flag = true; // прапор роботи циклу - піднято // Цикл вибору пунктів меню: while(flag) // поки піднято прапор { // Зчитуємо пункт меню у змінну: choice = menu_select(); // Перемикач вибору:</pre>
---	---

```

switch (choice)
{
    case 1: // якщо користувач ввів 1
        enter(invInfo); // вводим структуру
        break;
    case 2: // якщо користувач ввів 2
        list(invInfo); // виводимо всі структури
        break;
    case 3: // якщо користувач ввів 3
        del(invInfo); // очищуємо структуру
        break;
    case 4: // якщо користувач ввів 4
        flag=false; // опускаємо прапор циклу
        break; // трапиться вихід з циклу
    default: // якщо користувач ввів недійсний пункт
        cout << "Даний пункт відсутній,"
            << " спробуйте ще раз..." << endl;
}
}
return 0; // успішне закінчення програми
} // кінець функції main

// Функція зчитування пункту меню до змінної:
int menu_select()
{ // Змінна для збереження вибору користувача:
    int choice;
    // Запрошення на ввід і показ меню в консолі:
    cout << "Введіть Ваш вибір" << endl
        << "1)Введення елементів структури"<<endl
        << "2)Вивід всіх структур на екран"<<endl
        << "3)Очистка структури"<<endl
        << "4)Вихід"<<endl;
    cin >> choice; // вводим вибір користувача
    return choice; // повертаємо вибір з функції
}
// Функція введення елементів структури:
void enter( // опис формальних параметрів:
    inv *invInfo // масив структур для обробки
){
    // Описуємо змінну:
    int index; // індекс елемента

    // Запрошуємо на введення індексу:
    cout << "Введіть індекс елемента структури для введення: ";
    cin >> index; // вводим індекс

    // Перевірка потрапляння в межі масиву:
    while(index < 0 || index > size-1)
    { // Цикл не завершиться поки
        // користувач не введе коректний індекс,
        // повідомляємо про неправильне введення
    }
}

```

```

        cout << "Вихід за межі масиву! "
              << "Спробуйте ще раз!" << endl;
        cin >> index; // зчитуємо нове значення індексу
    }
    // Запрошуємо на введення і вводимо:
    cout << "Введіть назву елемента для інвентаризації: ";
    cin >> invInfo[index].item; // вводимо назву
    cout << "Введіть вартість елемента: ";
    cin >> invInfo[index].cost; // вводимо вартість
    cout << "Введіть кількість наявних елементів: ";
    cin >> invInfo[index].on_hand; // вводимо наявність
}
// Функція виводу масиву структур:
void list( // опис формальних параметрів:
    inv *invInfo // масив структур для обробки
){
    // Описуємо змінну:
    int nonzero = 0; // кількість ненульових структур

    // Організовуємо цикл за всіма структурами:
    for(int i = 0; i < size; i++)
    { // Виводимо тільки не порожні структури:
        if(invInfo[i].cost && invInfo[i].on_hand)
        { // Збільшуємо лічильник ненульових структур:
            nonzero++;
            // Вивід вмісту структури:
            cout << "Назва елемента: " << invInfo[i].item << endl;
            cout << "Ціна елемента: " << invInfo[i].cost << endl;
            cout << "Кількість: " << invInfo[i].on_hand << endl;
            cout << endl;
        }
    }
    // Якщо є не порожні структури, то виводимо
    // загальну кількість та кількість пустих структур
    if(nonzero)
    {
        cout << "Всього структур: " << size << endl;
        cout << "Порожніх структур: " << size-nonzero << endl;
    }
    else //если все структуры пустые выводим сообщение:
    {
        cout << "Доступно структур:" << size << endl;
    }
}
// Функція очищення масиву структур:
void del( // опис формальних параметрів:
    inv *invInfo // масив структур для обробки
){
    // Описуємо змінну:
    int index; // індекс елемента

```

	<pre>// Запрошуємо до введення: cout << "Введіть індекс елемента для видалення: "; cin >> index; // вводим індекса // Перевіряємо попадання в межі масиву: while(index < 0 index > size-1) { // Цикл не завершиться, поки користувач // не введе коректний індекс: cout << "Вихід за межі масиву! " << "Спробуйте ще раз!" << endl; cin >> index; // зчитуємо нове значення індексу } // Очищаємо поля структури: invInfo[index].cost = 0; // вартість елемента invInfo[index].on_hand = 0; // кількість елемента invInfo[index].item[0] = '\0'; // назва елемента // Повідомляємо про успішне видалення: cout << "Видалення " << index << " елемента пройшло успішно." << endl; }</pre>
С	<pre>Введіть Ваш вибір 1)Введення елементів структури 2)Вивід всіх структур на екран 3)Очистка структури 4)Вихід 1 Введіть індекс елемента структури для введення: 0 Введіть назву елемента для інвентаризації: Викрутка Введіть вартість елемента: 15 Введіть кількість наявних елементів: 30 Введіть Ваш вибір 1)Введення елементів структури 2)Вивід всіх структур на екран 3)Очистка структури 4)Вихід 2 Назва елемента: Викрутка Ціна елемента: 15 Кількість: 30 Всього структур: 4 Порожніх структур: 3 Введіть Ваш вибір 1)Введення елементів структури 2)Вивід всіх структур на екран 3)Очистка структури 4)Вихід</pre>

Завдання. Перевірити решту гілок програми.

Пам'ятайте! Відомий канадський вчений-інформатик Брайан Керніган сказав: «Процес підготовки програм для цифрового комп'ютера особливо привабливий не тільки тому, що він може бути економічно та науково корисним, а й тому, що це може бути естетичним досвідом, подібним до складання віршів чи музики».

Лабораторна робота №12

Тема: Робота з бінарними файлами в C++

Мета: освоєння роботи з бінарними файлами в C++.

Теоретична частина

12.1. Загальні поняття про бінарні файли

У лабораторній роботі №10 ми вже вивчили роботу з *текстовими файлами* або *файлами послідовного доступу*. Ці файли не придатні для використання у програмах миттєвого доступу, в яких необхідний запис повинен бути знайдений негайно. Типовими додатками миттєвого доступу є системи резервування авіаквитків, банківські системи, банкомати, мережеві магазини та інші системи обробки транзакцій, що вимагають швидкого доступу до конкретних даних. У банку можуть бути сотні тисяч (і навіть мільйони) інших клієнтів, і тим не менш, коли клієнт користується банкоматом, програма перевіряє стан його рахунку за лічені секунди. Такий спосіб миттєвого доступу стає можливим під час використання *бінарних файлів* або *файлів довільного доступу*. Доступ до окремих записів у файлі довільного доступу є безпосереднім (і швидким), а також не вимагає перегляду інших записів.

Дані у файл довільного доступу можна помістити без руйнування інших даних файлу. Раніше збережені дані також можна оновлювати або вилучати, не переписуючи весь існуючий файл.

12.2. Особливості роботи з бінарними файлами

Щоб отримати доступ до бінарного файлу з програми на C++, потрібно спочатку приєднати змінну потоку до файлу. Звичайні класи потоків `ofstream` (вихідний потік файлів) та `ifstream` (вхідний потік файлів) все ще є типами потоків, які потрібно використовувати. При цьому, забезпечується додатковий тип під назвою `fstream`, який дозволяє створювати файли з можливостями запису та читання з них.

Режими відкриття файлів наведені в таблиці 12.1.

Таблиця 12.1

Режими відкриття файлів в C++

Режим	Опис
<code>ios::in</code>	Відкрити файл для вводу.
<code>ios::out</code>	Відкрити файл для виводу.
<code>ios::app</code>	Дописати весь вивід в кінець файла.

Режим	Опис
<code>ios::ate</code>	Відкрити файл для виводу і переміститися в кінець файлу (зазвичай, застосовується для дописування даних в кінець файлу). Дані можуть бути записані в будь-яке місце файлу.
<code>ios::trunc</code>	Відкинути вміст файлу, якщо він існує (типово, це також робиться для <code>ios::out</code>).
<code>ios::binary</code>	Відкрити файл для двійкового (тобто нетекстового) вводу або виводу.

Приклад відкриття бінарного файлу.

```

. . .
#include <fstream> // підключаємо бібліотеку читання-запису файлів
. . .
    // Створюємо об'єкт ofstream з назвою infile:
    ifstream infile;
    // Елемент-функція open класу ofstream відкриває файл
    // з маршрутом E:\\hello.dat і прикріплює його до
    // існуючого об'єкта ofstream з назвою infile:
    infile.open("E:\\hello.dat", ios::binary | ios::in);
. . .

```

Нижче наведено приклад використання оператора `sizeof` для отримання розміру змінної для запису вмісту змінної на диск. Під час запису структури у файл ми копіюємо байти з оперативної пам'яті у файл!

Зверніть особливу увагу на використання структури.

```

. . .
// Підключаємо:
#include <fstream> // бібліотеку читання-запису файлів
. . .
struct Person // шаблон структури
{
    char name[50]; // рядок
    int age; // ціла змінна
    char phone[24]; // рядок
};
. . .
int main() // функція main виконується першою
{
    // Описуємо змінні:
    Person me = {"Robert", 28, "364-2534"}; // структура
    Person book[30]; // масив структур
    int x = 123; // ціла змінна
    double fx = 34.54; // дійсна змінна

    ofstream outfile; // створюємо об'єкт для запису

```

```

// Створюємо бінарний файл junk.dat у поточному каталозі:
outfile.open("junk.dat", ios::binary | ios::out);
// Записуємо дані у бінарний файл – sizeof() визначає
// розмір об'єктів, які записуються:
outfile.write((char*)&x, sizeof(int)); // записуємо цілу змінну
outfile.write((char*)&fx, sizeof(fx)); // записуємо дійсну змінну
outfile.write((char*)&me, sizeof(me)); // записуємо структуру
outfile.write((char*)&book, 30*sizeof(Person)) // записуємо
// масив структур

outfile.close(); // закриваємо файл
. . .
return 0; // успішне закінчення програми
} // кінець функції main

```

Механізм запису в бінарний файл налаштований таким чином, що для кожної змінної потрібно *приводити її адресу до типу char** виразом **(char*)&address**.

Читання даних з двійкового файлу виконується так само, як і їх запис, за винятком того, що функція тепер називається `read`, а не `wright`. Під час читання даних ми копіюємо *байти з файлу у оперативну пам'ять!* Зверніть увагу на розмір даних, які зчитуються.

```

. . .
#include <fstream> // підключаємо бібліотеку читання-запису файлів
. . .
int main() // функція main виконується першою
{
    // Описуємо змінну:
    int x; // ціла змінна 4 байти

    ifstream infile; // створюємо об'єкт для читання
    // Відкриваємо бінарний файл silly.dat у поточному каталозі:
    infile.open("silly.dat", ios::binary | ios::in)
    infile.read(&x, 4); // зчитуємо з файлу цілу змінну
    // sizeof(int) = 4
    infile.close(); // закриваємо файл
. . .
return 0; // успішне закінчення програми
} // кінець функції main

```

12.3. Завдання до роботи з бінарними файлами

Кожен студент вибирає варіант завдання за номером у списку групи.

Завдання: знайти рішення задачі відповідно до номера варіанту.

Підготувати звіт з виконаної роботи, який повинен містити:

– номер, тему і мету лабораторної роботи;

- варіант роботи відповідно до номера студента у списку;
- текст програми мовою C++ з коментарями;
- результати роботи програми;
- висновки до лабораторної роботи.

№	Варіант
1.	<p>Дано інформацію про товари, що зберігаються на складі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Найменування товару. ➤ Вартість товару. ➤ Країна-виробник товару. ➤ Кінцевий термін реалізації товару. ➤ Кількість, наявного товару. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
2.	<p>Дано інформацію про жорсткі диски. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Ємність накопичувача. ➤ Тип жорсткого диска. ➤ Форм-фактор. ➤ Інтерфейс. ➤ Швидкість обертання шпинделя. ➤ Ціна. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
3.	<p>Дано інформацію про країни світу. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва країни. ➤ Площа країни. ➤ Населення країни. ➤ Державна мова. ➤ Грошова одиниця. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
4.	<p>Дано інформацію про продукти харчування. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва продукту. ➤ Калорійність продукту. ➤ Вміст у продукті вітамінів (А, В, С, Е, Р, РР). ➤ Вартість 1 кг продукту. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>

№	Варіант
5.	<p>Дано інформацію про потяги. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Номер поїзда. ➤ Станція відправлення. ➤ Станція призначення. ➤ Час відправлення. ➤ Час прибуття. ➤ Тип поїзда (пасажирський, швидкий). <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
6.	<p>Дано інформацію про книги. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Автор. ➤ Назва. ➤ Видавництво. ➤ Рік видання. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
7.	<p>Дано інформацію про пасажирів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ ПІБ пасажирів. ➤ Номер рейсу. ➤ Реєстраційний номер багажу. ➤ Вага багажу. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
8.	<p>Дано інформацію про студентів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ ПІБ студента. ➤ Група. ➤ Пол. ➤ Екзаменаційні оцінки з вищої математики та фізики. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
9.	<p>Дано інформацію про співробітників фірми. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ ПІБ співробітника. ➤ Рік народження. ➤ Пол. ➤ Посада. ➤ Термін роботи у фірмі. ➤ Заробітна плата. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>

№	Варіант
10.	<p>Дано інформацію про автомобілі, що здаються на прокат. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Марка автомобіля. ➤ Країна-виробник. ➤ Рік випуску. ➤ Довжина пробігу. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
11.	<p>Дано інформацію про смартфони. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва смартфона. ➤ Фірма-виробник. ➤ Діагональ екрану. ➤ Оперативна пам'ять. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
12.	<p>Дано інформацію про твори мистецтва, виставлені на аукціонний продаж. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Номер лота. ➤ Назва твору. ➤ Автор. ➤ Країна. ➤ Початкова вартість. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
13.	<p>Дано інформацію про каталог всіх книг, видатних на абонементі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Автор (и). ➤ Назва. ➤ Обсяг (у сторінках). ➤ Кількість примірників бібліотеки. ➤ "Жанр" (навчальні посібники, класична художня література, сучасний детектив, сучасна фантастика). <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
14.	<p>Дано інформацію про тварин, представлених на виставку котів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Кличка. ➤ Колір. ➤ Порода. ➤ Вік. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>

№	Варіант
15.	<p>Дано інформацію про проведення вітрильної регати. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Назва яхти. ➤ Клас яхти (Фінн, Зоряний, Торнадо, Солінг). ➤ Порт приписки. ➤ Зайняте місце на регаті. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
16.	<p>Дано інформацію про замовлення, що надходять до годинної майстерні. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Номер замовлення. ➤ Прізвище замовника. ➤ Марка годин. ➤ Термін виконання. ➤ Вартість замовлення. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
17.	<p>Дано інформацію про ноутбуки. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Фірма-виробник. ➤ Тактова частота процесора. ➤ Діагональ екрану. ➤ Вартість. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
18.	<p>Дано інформацію про пропоновані до продажу автомобілі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Країна-виробник. ➤ Марка автомобіля. ➤ Колір корпусу. ➤ Ціна автомобіля. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
19.	<p>Дано інформацію про планшети. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Розмір планшета. ➤ Вага планшета. ➤ Ціна планшета. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>

№	Варіант
20.	<p>Дано інформацію про планшети. про фотоапарати. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Зум. ➤ Матриця (у пікселях). ➤ Ціна. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
21.	<p>Дано інформацію для обробки інформації про комп'ютерні манипулятори тапу «миша». Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Тип миші. ➤ Роз "єм для підключення. ➤ Чутливість. ➤ Кількість кнопок. ➤ Ціна. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
22.	<p>Дано інформацію для обробки інформації про відеореєстраторів. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Максимальна роздільна здатність відео. ➤ Вбудований GPS. ➤ Роздільна здатність сенсора. ➤ Ціна. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
23.	<p>Дано інформацію про відеокарти. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Графічний чіп. ➤ Частота пам 'яті. ➤ Обсяг пам 'яті. ➤ Тип пам 'яті. ➤ Максимально підтримувана роздільна здатність. ➤ Ціна. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>

№	Варіант
24.	<p>Дано інформацію про телевізори. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Компанія-виробник. ➤ Діагональ екрану. ➤ Підтримка Smart TV. ➤ Дозвіл. ➤ Wi-Fi. ➤ Ціна. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
25.	<p>Дано інформацію для ведення обліку торгових операцій фірми. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ Реєстраційний номер операції. ➤ Сутність угоди (товар). ➤ Обсяг угоди (грн.). ➤ Форма розрахунку (нал., безнал., бартер). ➤ Дата вчинення. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>

12.4. Приклад розв'язання завдання до лабораторної роботи

	<p>Дано інформацію з інвентаризації товарів, що зберігаються на складі. Інформація включає такі дані:</p> <ul style="list-style-type: none"> ➤ назва елемента; ➤ вартість елемента; ➤ кількість наявних елементів. <p>Створити відповідну структуру. Записати її у бінарний файл та прочитати.</p>
--	---

Нижче наведено приклад розв'язання завдання до лабораторної роботи.

➡	<pre>// Підключаємо: #include <iostream> // бібліотеку вводу-виводу #include <windows.h> // заголовний файл локалізації #include <string.h> // бібліотеку з ф-ціями роботи з рядками #include <fstream> // бібліотеку читання-запису файлів using namespace std; // Шаблон структури Інвентаризація:</pre>
---	--

```

struct Inventory {
    char title[25]; //назва елемента
    double price;    // вартість елемента
    int quantity;    // кількість наявних елементів
};

int main() // функція main виконується першою
{
    SetConsoleCP(1251);        // локалізуємо ввід
    SetConsoleOutputCP(1251);  // локалізуємо вивід

    Inventory invSpec; // структура для запису у файл
    Inventory w1;       // структура для читання з файлу

    // Заповнюємо структуру:
    strncpy(invSpec.title,    // текстове поле заповнюємо за допо-
        "Викрутка", 25); // могою рядкової функції копіювання
    invSpec.price = 23.59;    // вартість елемента
    invSpec.quantity = 53;    // кількість наявних елементів


    // Створюємо об'єкт f1 для запису у файл Invent.bin:
    ofstream f1("G:\\Lab_11\\Invent.bin", ios::binary | ios::out);
    // Записуємо структуру у файл:
    f1.write((char*)&invSpec, sizeof(invSpec));
    // Закриваємо файл:
    f1.close();

    // Створюємо об'єкт f2 для читання з файлу Invent.bin:
    ifstream f2("G:\\Lab_11\\Invent.bin", ios::binary | ios::in);
    // Читаємо структуру з файлу у w1:
    f2.read((char*)&w1, sizeof(invSpec));
    // Закриваємо файл:
    f2.close();

    // Виводимо дані на екран:
    cout << "\n Назва: " << w1.title
        << "   Ціна: " << w1.price
        << "   Кількість: " << w1.quantity << endl;

    return 0;    // успішне закінчення програми
}               // кінець функції main

```

	Назва: Викрутка Ціна: 23.59 Кількість: 53

Зверніть увагу, що на диску G:\ у каталозі Lab_11 з'явився файл Invent.bin (рис. 21.1).

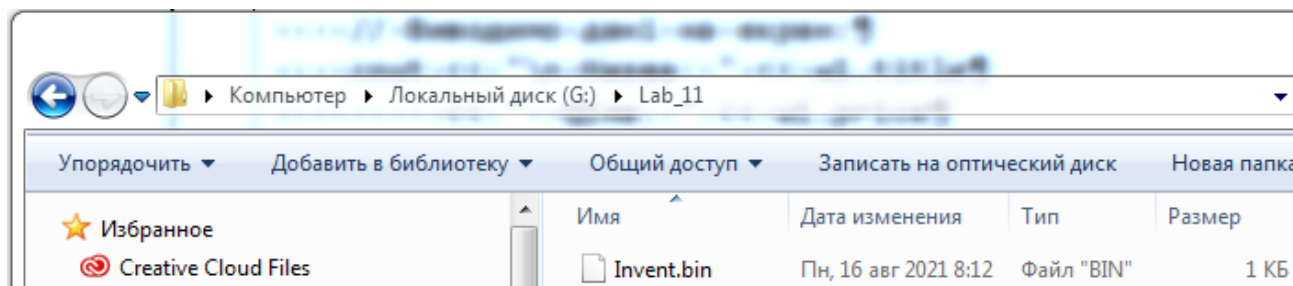


Рис. 11.1. Файл Invent.bin на диску G:\

Пам'ятайте! Відомий канадський вчений-інформатик Брайан Керніган сказав: «Я хочу, щоб студенти розуміли конкретні технології, але справжня мета полягає в тому, щоб вони могли міркувати про те, як працюють системи, і ставитися до технологій з розумним скептицизмом, щоб, коли через декілька років вони будуть керувати світом, вони добре робили свою справу».

**Міністерство освіти і науки України
НТУ «Дніпровська політехніка»**



**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних технологій та комп'ютерної інженерії**

**ЗВІТ
про виконання лабораторних робіт
з дисципліни
«Обчислювальна техніка та програмування»**

На тему: «Основи роботи з числовими типами даних в C++»

Виконав:
студент(ка) гр. _____

(П.І.Б.)

Прийняв:
викладач. каф. ІТКІ

(П.І.Б.)

**Дніпро
2021**

Підготовка до лабораторних робіт на C++

Тема: Встановлення і запуск Code::Blocks

Мета: ознайомитися з встановленням і запуском Code::Blocks.

Теоретична частина

1. Визначення функцій Code::Blocks

Code::Blocks – вільне інтегроване середовище розробки (IDE – Integrated Development Environment) з відкритим вихідним кодом, яка підтримує використання різних компіляторів. Типово, Code::Blocks використовує компілятор MinGW, який поставляється в комплекті (можна завантажити дистрибутив і без вбудованого компілятора). MinGW – це варіант компілятора GNU C/C++ для Windows. MinGW – це скорочення «Minimalist GNU для Windows».

Набір MinGW включає інструменти для програмування з відкритим вихідним кодом, які не залежать від будь-яких платних сторонніх бібліотек. Він використовується багатьма середовищами розробки (Visual Studio використовує власний компілятор). MinGW включає набір GNU Compiler Collection (GCC), включаючи компілятори C, C++, ADA і Fortran. Code::Blocks має відкриту архітектуру, що дозволяє йому масштабуватися за рахунок модулів (можна писати програми як під Windows, Linux, так і під різні мікропроцесори типу avr і stm32). Відлагодник коду в Code::Blocks підтримує точки зупину у вихідному коді або в даних, які обробляє програма.

2. Встановлення компілятора MinGW

Встановлення самого Code::Blocks не представляє нічого складного. Бажано тільки встановлювати його в теку, в маршруті до якої немає пробілів або кирилиці. Дистрибутив Code::Blocks версії 17.12 містив стару версію компілятора MinGW, тому доводилося окремо завантажувати Code::Blocks без компілятора MinGW і потім встановлювати компілятор. Зараз версія Code::Blocks 20.03 містить актуальну версію MinGW (за замовчуванням використовується 64-розрядна версія компілятора, але можна завантажити дистрибутив і з 32-розрядною версією). Але про всяк випадок розглянемо окремо установку MinGW (на прикладі 32-розрядної версії). Спочатку скачуємо дистрибутив Code::Blocks з сайту проекту mingw.org. Після запуску з'явиться наступне вікно (рис. 1).

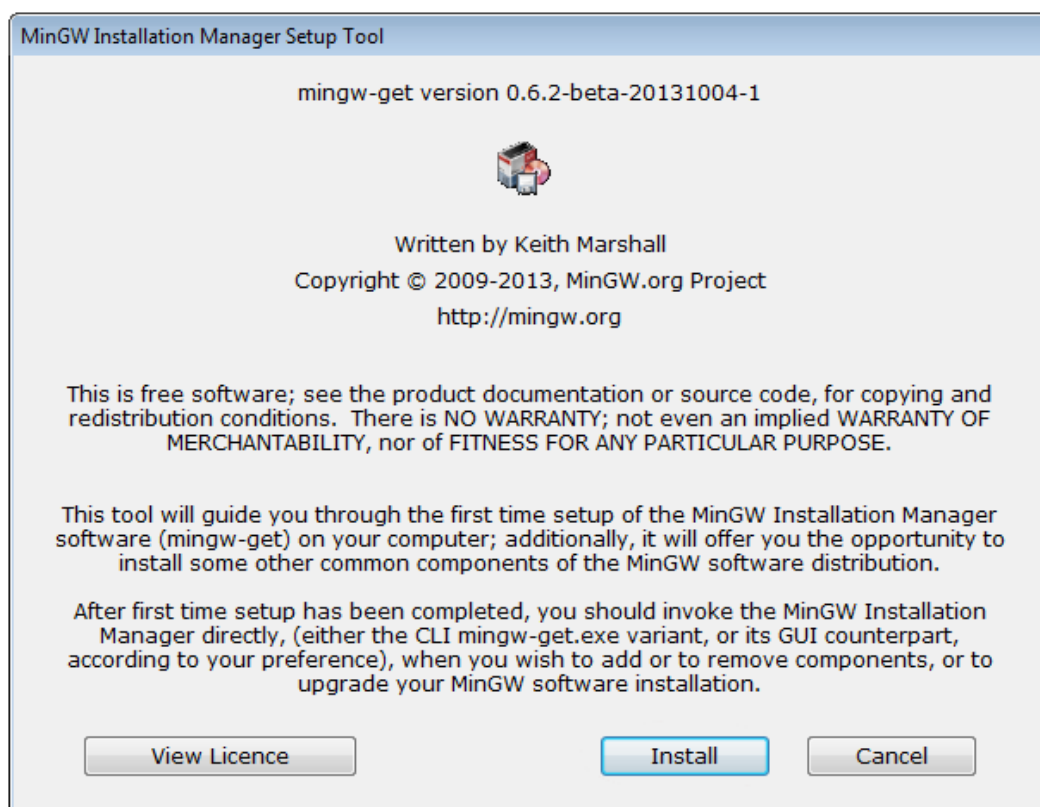


Рис. 1. Запуск дистрибутива Code::Blocks

Натискаємо Install і отримуємо наступне вікно (рис.2).

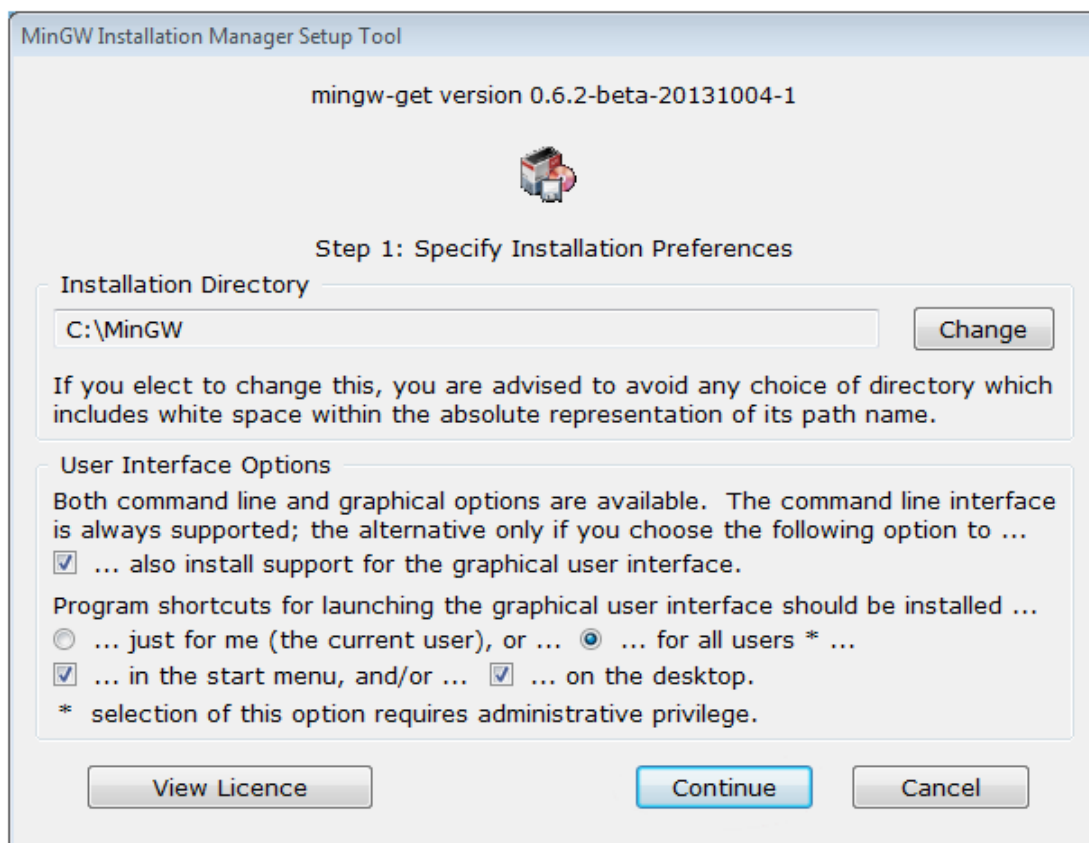


Рис. 2. Наступне вікно

Тут у полі **Installation Directory** вказуємо розташування файлів компілятора. Типовим є `C:\MinGW`, його і залишаємо. Натискаємо **Continue** і починається завантаження і встановлення менеджера для встановлення MinGW.

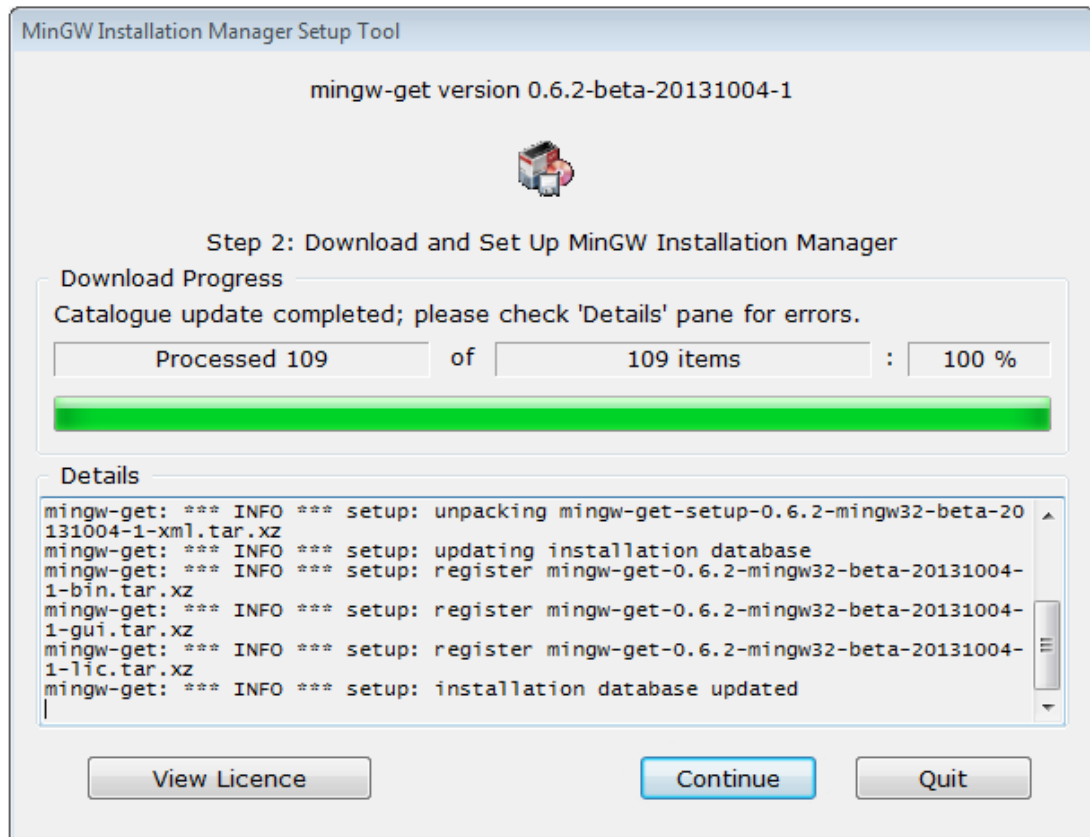


Рис. 3. Завантаження і встановлення менеджера для встановлення MinGW

Натискаємо **Continue** і отримуємо наступне вікно (рис. 4):

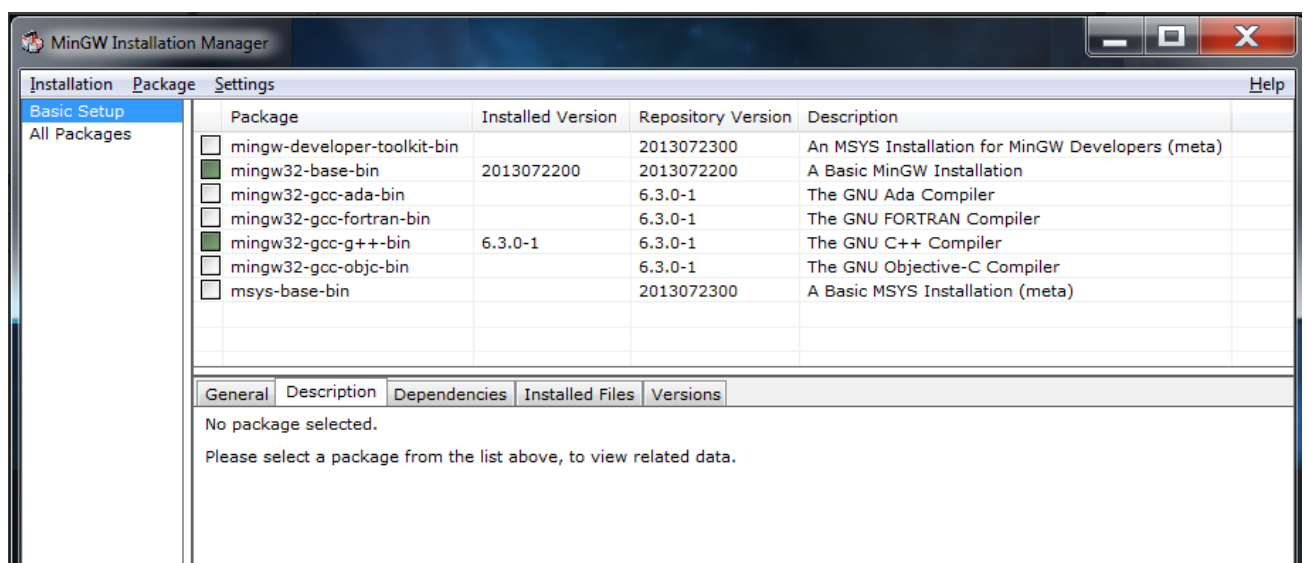


Рис. 4. Наступне вікно

Для програмування на C++ вибираємо для встановлення пакети mingw32-base-bin (A Basic MinGW Installation) і mingw32-gccg++-bin (The GNU C++ Compiler), а потім тиснемо в меню Installation→Apply Changes. З'явиться вікно, в якому буде показано завантаження вибраних пакунків (рис.5).

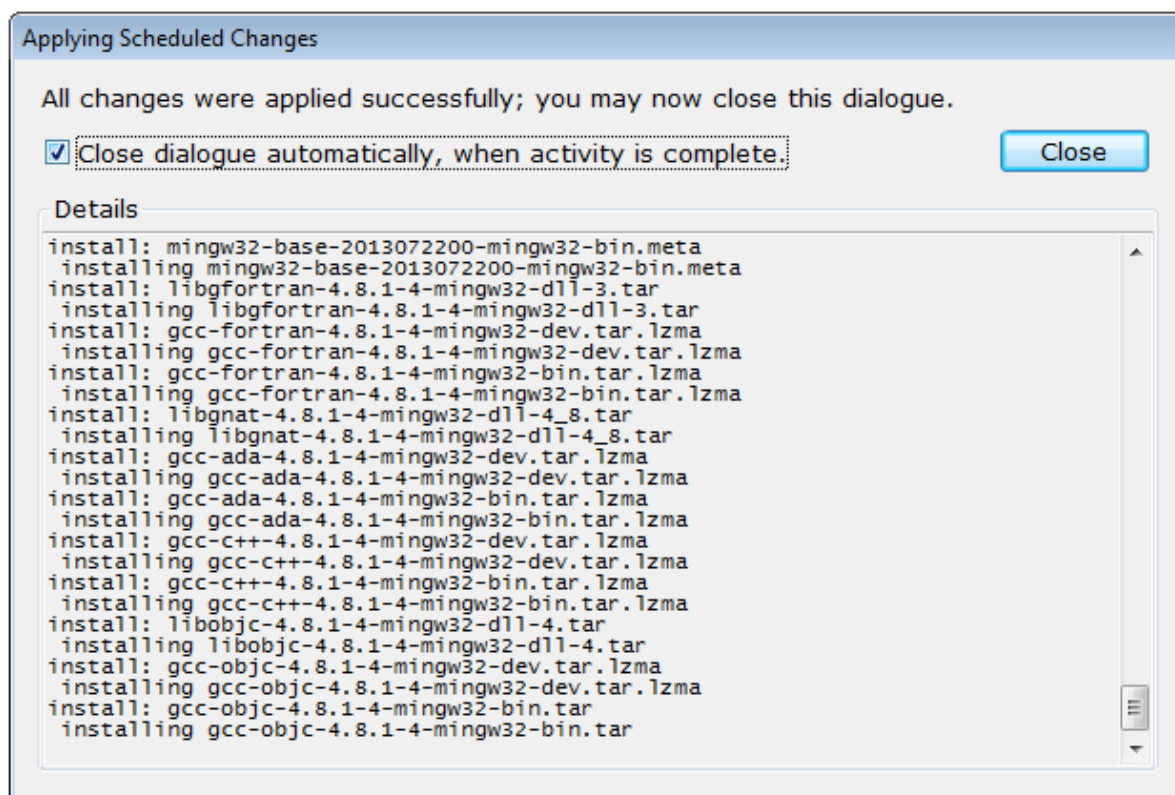


Рис. 5. Завантаження вибраних пакунків

Після завершення звантаження пакунків натискаємо на кнопку Close.

Під час першого запуску Code::Blocks автоматично виявить встановлений компілятор. При виникненні проблем необхідно перейти в меню Settings→Compiler... і на вкладці Toolchain executables перевірити правильність шляху до MinGW, а також імена файлів компілятора і лінковика. Тепер можна створювати проект.

3. Створення проекту

Щоб створити проект, вам слід перейти до меню File→New→Project.... Або відразу натиснути кнопку Create a new project на стартовій сторінці (рис. 6).

У вікні, що з'явиться, вибираємо потрібний тип проекту. Тут ми бачимо, що Code::Blocks має велику кількість шаблонів проектів. У нашому випадку це Console application (консольна програма) (рис. 7).

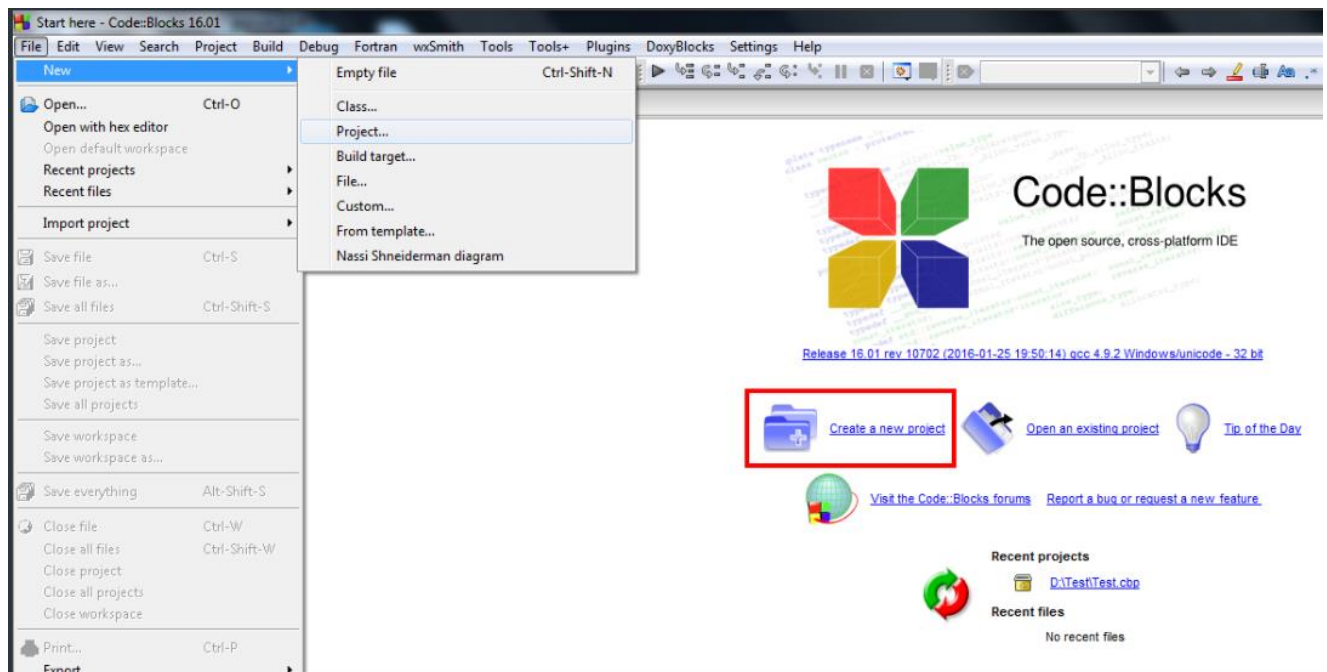


Рис. 6. Стартова сторінка Code::Blocks

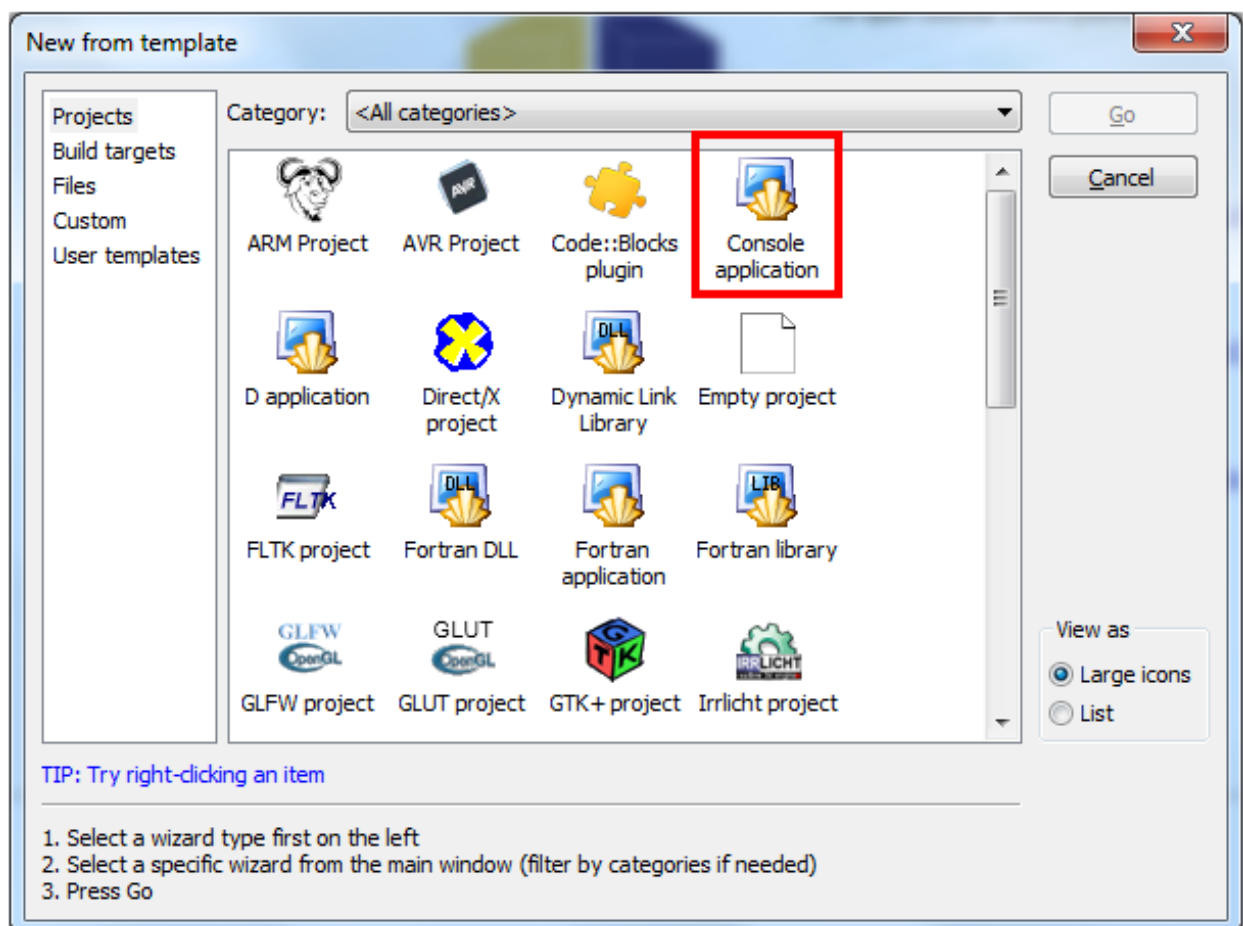


Рис. 7. Вибираємо потрібний тип проекту – Console application

Потім клацаємо кнопку Go і переходимо до наступного вікна. У наступному вікні обираємо мову програмування. Вибираємо C++ і натискаємо на кнопку Next (рис. 8).

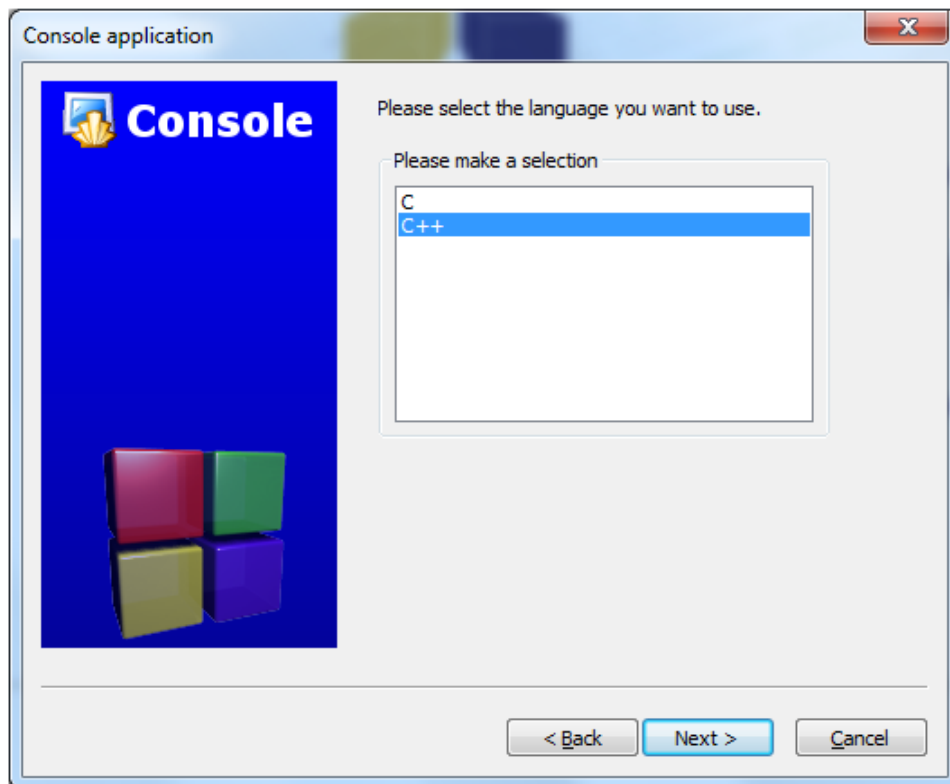


Рис. 8. Вибір мови програмування

Далі в полі **Project Title** вписуємо назву проекту. У нашому випадку це **Test**. Тут же вказуємо розташування проекту на диску (поле **Folder to create project in**). У вказаній нами теці **Code::Blocks** автоматично створить теку з раніше вказаною назвою проекту. Вказавши назву теки і назву файла, натискаємо кнопку **Next** (рис. 9).

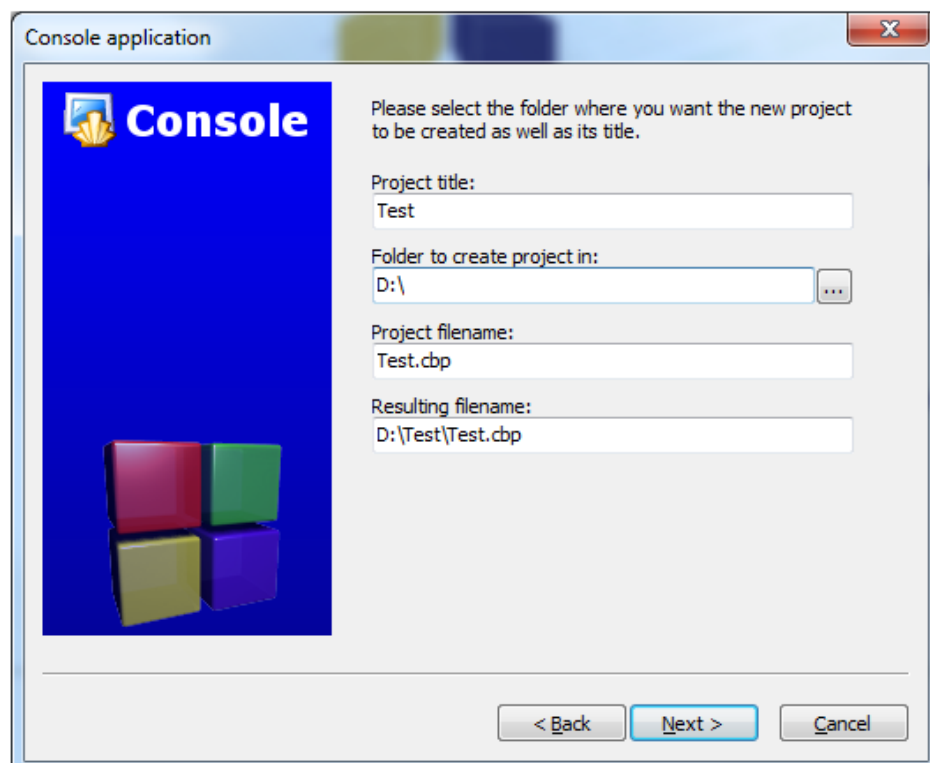


Рис. 9. Визначання проекту

У наступному вікні вибираємо компілятор, а також скрипти збирання. За замовчуванням вибрано **GNU GCC Compiler**, який ми і будемо використовувати. Скрипти збірки дозволяють отримати кілька версій однієї програми. Типовим є два скрипти: **Debug** (використовується при відладці) і **Release** (сценарій компіляції готової програми). Хоча б один зі сценаріїв повинен бути обраний. Якщо вибрано обидва сценарії, ви зможете перемикатися між ними (рис.10).

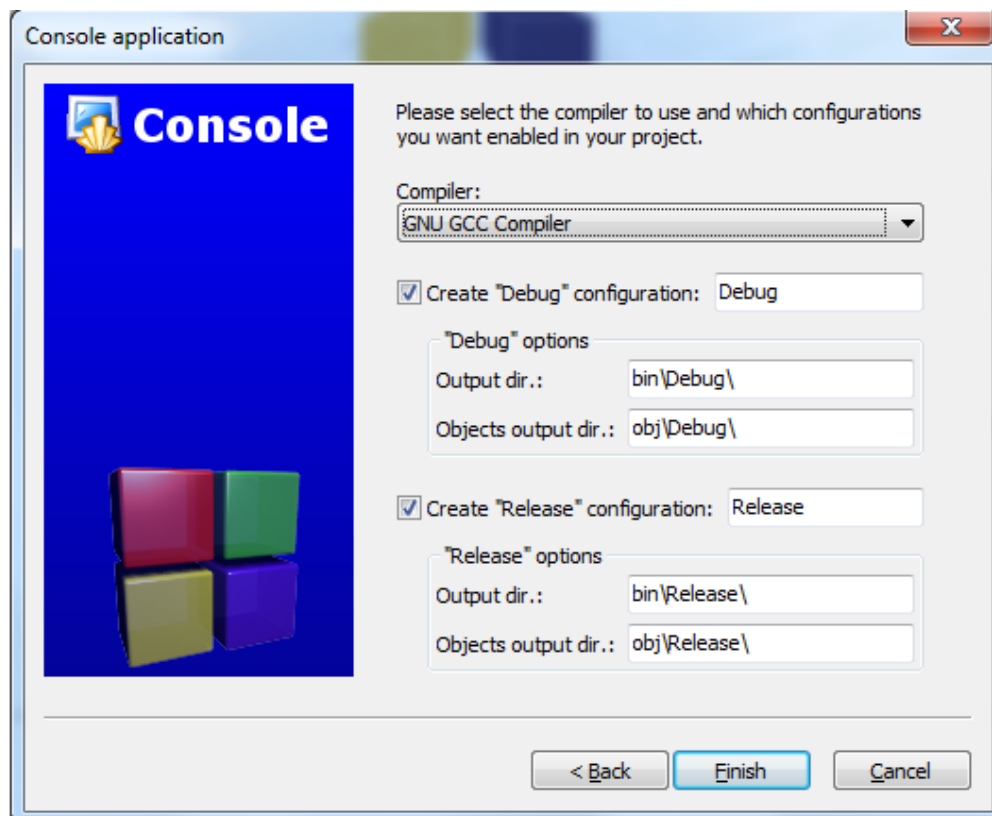


Рис. 10. Вибір компілятора і скриптів збирання

Тиснемо **Finish** і отримуємо мінімальну програму, так званий **Hello World!** (рис. 11).

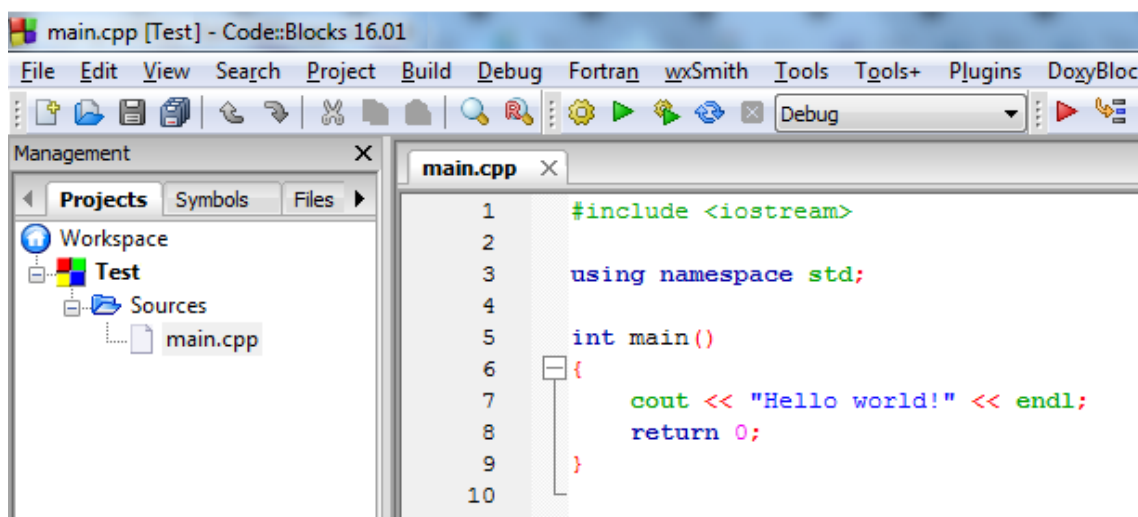


Рис. 11. Програма Hello World!

Натискаємо F9 для компіляції та запуску програми і отримуємо наступне вікно (рис. 12).

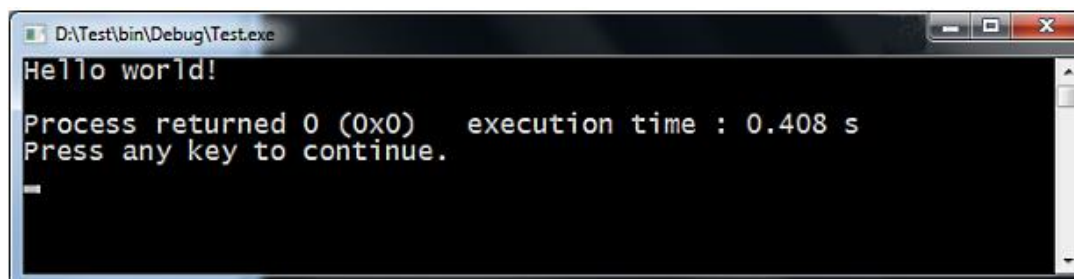


Рис. 12. Результат роботи програми

Зверніть увагу на кнопки на панелі інструментів, призначені для запуску та компіляції програми (рис. 13).



Рис. 13. Кнопки, призначені для запуску та компіляції програми

Перша ліворуч кнопка (з жовтою шестерінкою) запускає тільки компіляцію програми, друга (із зеленим трикутником) запускає на виконання останній скомпільований файл програми, третя (із зеленим трикутником і жовтою шестірнею) – спочатку запускає компіляцію програми, а потім запускає отриманий ехе-файл програми.

Частою помилкою є використання кнопки запуску (з зеленим трикутником) для компіляції і запуску програми. У цьому випадку внесені до вихідного коду зміни не впливатимуть на поведінку програми, оскільки програма не була перекомпільована. Тому краще завжди використовувати третю кнопку (із зеленим трикутником і жовтою шестірнею).

Коди ASCII (0 – 127)

(American Standard Code for Information Interchange)

DEC	CHAR	Name	DEC	CHAR	DEC	CHAR	DEC	CHAR
0	Ctrl-@	NUL	32	SPC	64	@	96	'
1	Ctrl-A	BOH	33	!	65	A	97	a
2	Ctrl-B	STX	34	"	66	B	98	b
3	Ctrl-C	ETX	35	#	67	C	99	c
4	Ctrl-D	EOT	36	\$	68	D	100	d
5	Ctrl-E	ENQ	37	%	69	E	101	e
6	Ctrl-F	ACK	38	&	70	F	102	f
7	Ctrl-G	BEL	39	'	71	G	103	g
8	Ctrl-H	BS	40	(72	H	104	h
9	Ctrl-I	HT	41)	73	I	105	i
10	Ctrl-J	LF	42	*	74	J	106	j
11	Ctrl-K	VT	43	+	75	K	107	k
12	Ctrl-L	PF	44	,	76	L	108	l
13	Ctrl-M	CR	45	-	77	M	109	m
14	Ctrl-N	SO	46	.	78	N	110	n
15	Ctrl-O	SI	47	/	79	O	111	o
16	Ctrl-P	DLE	48	0	80	P	112	p
17	Ctrl-Q	DC1	49	1	81	Q	113	q
18	Ctrl-R	DC2	50	2	82	R	114	r
19	Ctrl-S	DC3	51	3	83	S	115	s
20	Ctrl-T	DC4	52	4	84	T	116	t
21	Ctrl-U	NAK	53	5	85	U	117	u
22	Ctrl-V	SYN	54	6	86	V	118	v
23	Ctrl-W	ETB	55	7	87	W	119	w
24	Ctrl-X	CAN	56	8	88	X	120	x
25	Ctrl-Y	EM	57	9	89	Y	121	y
26	Ctrl-Z	SUB	58	:	90	Z	122	z
27	Ctrl-[ESC	59	;	91	[123	(
28	Ctrl-\	FS	60	<	92	\	124	
29	Ctrl-]	GS	61	=	93]	125)
30	Ctrl-^	RS	62	>	94	^	126	~
31	Ctrl-_	US	63	?	95	_	127	DEL

Список рекомендованої літератури

1. Allibang S. C++ Programming Language: Simple, Short, and Straightforward Way of Learning C++ Programming. – New York: DBest Reads, 2016. – 218 p.
2. Andrist B., Sehr V. C++ High Performance: Master the art of optimizing the functioning of your C++ code. – 2nd ed. – Birmingham: Packt Publishing, 2020. – 540 p.
3. Anggoro Wisnu, Torjo John. Boost.Asio C++ Network Programming. – Packt Publishing, 2015. – 200 p.
4. Anggoro Wisnu. C++ Data Structures and Algorithms. – Packt Publishing, 2018. – 322 p.
5. Bancila M. The Modern C++ Challenge. – Packt Publishing, 2018. – 328 p.
6. Bancila Marius. Modern C++ Programming Cookbook. – Packt Publishing, 2017. – 590 p.
7. Bjornander S. C++ Windows Programming. – Birmingham: Packt Publishing, 2016. – 577 p.
8. Bjornander Stefan. C++17 By Example. – Packt Publishing, 2018. – 442 p.
9. Carney Ben. C++ High Performance. – Packt Publishing, 2018. – 318 p.
10. Dale Nell, Weems Chip, Richards Tim. C++ Plus Data Structures. – 6th Edition. – Jones & Bartlett Learning, 2018. – 1345 p.
11. Deitel Paul, Deitel Harvey. C++20 for Programmers. 3rd Edition. – Pearson Education, 2020. – 291 p.
12. Deitel Paul, Deitel Harvey. C++ How to program. Tenth Edition. – Pearson Education, 2017. – 1075 p.
13. Dmitrovic Slobodan. Modern C++ for Absolute Beginners: A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards. – Apress, 2020. – 323 p.
14. Etter D.M., Ingber J.A. Engineering Problem Solving With C++. – 4th ed. – Pearson, 2017. – 702 p.
15. Filipek Bartłomiej. C++17 in Detail. Learn the Exciting Features of The New C++ Standard. – Leanpub, 2019. – 357 p.
16. Gaddis T. Starting Out with C++ from Control Structures to Objects. – 9th ed. – Pearson, 2017. – 1344 p.
17. Gaddis Tony, Walters Judy, Muganda Godfrey. Starting Out with C++: Early Objects. – 10th Edition. – Pearson Education, Inc., 2020. – 1376 p.
18. Galowicz Jacek. C++17 STL Cookbook. – Packt Publishing, 2017. – 532 p.
19. Green D. The C++ Workshop: A New, Interactive Approach to Learning C++. – Packt, 2020. – 598 p.
20. Gregoire M. Professional C++. – 4th ed. – Wrox, 2018. – 1176 p.

21. Grimm R. Concurrency with Modern C++. – LeanPub, 2019. – 474 p.
22. Grose Dontae M. Programming in C++. – Amazon Digital Services, 2017. – 121 p.
23. Halterman Richard L. Fundamentals of C++ Programming. – Collegedale, Tennessee (United States of America), School of Computing Southern Adventist University, 2018. – 766 p.
24. Horton Ivor, Van Weert Peter. Beginning C++ 20: From Novice to Professional. – Apress Media LLC., 2020. – 825 p.
25. Jason Turner. C++ Best Practices. – Lenpab, 2020. – 127 p.
26. Josuttis N.M. C++17. The Complete Guide. – Lean Publishing, 2019. – 370 p.
27. Kormanyos Christopher. Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming. – Springer, 2018. – 428 p.
28. Lippman S., Lajoie J., Moo B. C++ Primer. – Fifth Edition. – Addison–Wesley, 2017. – 963 p.
29. Lischner Ray. Exploring C++20 The Programmer's Introduction to C++. – 3rd Edition. – Apress, 2020. – 653 p.
30. Stroustrup Bjarne. The C++ Programming Language. – 4th edition. – Addison–Wesley Professional, 2013. – 1206 p.
31. Stroustrup B. Programming: Principles and Practice Using C++. – 2nd Edition. – Addison–Wesley Professional, 2014. – 1312 p.
32. Stroustrup B. A Tour of C++. – 2nd Edition. – Addison–Wesley Professional, 2018. – 256 p.
33. Вступ до програмування мовою C++. Організація обчислень : навч. посіб. / Ю. А. Бєлов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К. : Видавничо–поліграфічний центр "Київський університет", 2012. – 175 с.
34. Грицюк Ю., Рак Т. Програмування мовою C++. Навчальний посібник. – Львів: Вид–во ЛДУ БЖД, 2011. – 146 с.
35. Зубенко В.В., Омельчук Л.Л. Програмування. Поглиблений курс. – К.:Видавничо–поліграфічний центр “Київський університет”, 2011. – 623 с.
36. C++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2–ге вид. перероб. і доповн. – Одеса : Фенікс, 2019. – 477 с.