

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

Кафедра електронної техніки

**МЕТОДИЧНІ ВКАЗІВКИ
до виконання практичних занять
з дисципліни «Комп'ютерні технології та
програмування. Частина 1»**

для студентів денної та заочної форм навчання
всіх спеціальностей

Луцьк, 2023

УДК 004.4 (072)
М 54

Методичні вказівки до виконання практичних занять з дисципліни «Комп'ютерні технології та програмування. Частина 1» : для студ. денної та заочної форм навч. всіх спец. [Електронний ресурс] / уклад. : Т. В. Алтухова, Е. А. Петелін. – Луцьк : ДВНЗ «ДонНТУ», 2023. – 39 с.

Методичні вказівки до виконання практичних занять з дисципліни «Комп'ютерні технології та програмування. Частина 1» містять необхідні положення для підготовки та виконання практичних занять, викладені завдання та приклади їх вирішення, основні вимоги до змісту звіту та питання до самоконтролю. Завдання виконуються студентами у відповідності до заданого варіанту.

Укладачі: Алтухова Т. В., канд. техн. наук, доцент кафедри електронної техніки
Петелін Е. А., канд. техн. наук, доцент, доцент кафедри електронної
техніки

Рецензент: Калиниченко В. В., к.т.н., доцент кафедри прикладної механіки

Відповідальний за випуск: Вовна О. В., д-р. техн. наук, проф., зав. кафедри
електронної техніки ДВНЗ «ДонНТУ»

Затверджено навчально-методичним відділом ДонНТУ,
протокол № 9 від 27.06.2023ю р.

Розглянуто на засіданні кафедри електронної техніки ДВНЗ «ДонНТУ»
протокол № 12 від 20.06.2023 р.

ЗМІСТ

ВСТУП	4
Практична робота №1 СКЛАДАННЯ АЛГОРИТМІВ РОЗГАЛУЖЕНОЇ ТА ЦИКЛІЧНОЇ СТРУКТУР	5
Короткі теоретичні відомості	5
Завдання до практичної роботи №1	9
Контрольні запитання.....	15
Практична робота №2 ОСНОВИ ПРОГРАМУВАННЯ В ПРОГРАМНОМУ СЕРЕДОВИЩІ DEV-C++	16
Короткі теоретичні відомості	16
Завдання до практичної роботи №2	20
Контрольні запитання.....	22
Практична робота №3 ОРГАНІЗАЦІЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ.....	23
Короткі теоретичні відомості	23
Завдання до практичної роботи №3	26
Контрольні запитання.....	28
Практична робота №4 УМОВНІ ОПЕРАТОРИ ТА ОПЕРАТОРИ ВИБОРУ	29
Короткі теоретичні відомості	29
Завдання до практичної роботи №4	32
Контрольні запитання.....	37
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	38

ВСТУП

На сьогоднішній день однією з розповсюджених мов програмування є мова C/C++, на основі якої зараз формуються багато новітніх мов програмування та програмних середовищ. Саме її властивості сприяли такому розвитку, а саме лаконічність, потужність, гнучкість, мобільність, можливість доступу до всіх функціональних засобів системи. Програмувати на C++ можна як для Windows, так і для Unix, причому для кожної з операційних систем існує значна кількість засобів розробляння: від компіляторів до потужних інтерактивних середовищ, прикладом яких є Dev-C++, що розглядатиметься в даних методичних вказівках.

В методичних вказівках містяться необхідні положення щодо підготовки та виконання практичних занять, наводяться завдання та приклади їх вирішення, що дозволять опонувати базовим елементам алгоритмізації та основ програмування на мові C/C++. Завдання виконуються студентами у відповідності до заданого варіанту, яке узгоджується з викладачем. Після виконання практичних робіт студент повинен оформити звіт згідно наведених вимог до кожної з них.

На титульному аркуші звітів з практичних робіт обов'язково вказується назва дисципліни, тема роботи та автор звіту.

Практична робота №1

**СКЛАДАННЯ АЛГОРИТМІВ РОЗГАЛУЖЕНОЇ ТА ЦИКЛІЧНОЇ
СТРУКТУР**

Мета: набути навички побудови алгоритмів розв’язування задач та надання їх схематичного відображення у вигляді блок-схем

Короткі теоретичні відомості

Алгоритми вирішення складних задач передбачає певну систему правил, які визначає послідовність відповідних операцій, що дозволяють отримати остаточний результат розв’язку.

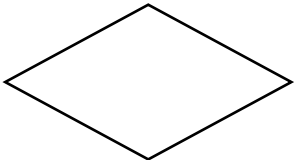
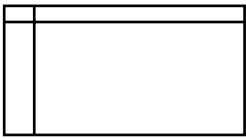
Алгоритм – це кінцева сукупність чітко визначених правил розв’язку деякого класу задач або відповідний набір інструкцій, що дозволяє описати певний порядок дій виконавця для вирішення певного завдання [1].

Блок-схема – це графічне відображення алгоритмів, що містить геометричні фігури, які сполучені лініями зі стрілками поміж собою. Геометричні фігури визначають тип дії, а вже текст в середині цих блоків пояснює кожну дію окремо. Стрілки показують напрямок виконання послідовності команд, що визначені алгоритмом.

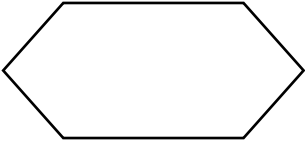
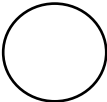
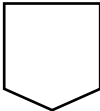
Геометричні фігури блок-схем, а також їх розміри та товщина ліній і кути нахилу відображуються у відповідності до державних стандартів (ISO 5807:2016) [1]. Самі ж алгоритми можуть бути як у спрощеному вигляді, так і мати більш детальний опис. Нижче наведені основні символи, що використовуються при побудові різноманітних алгоритмів (табл. 1.1).

Існує декілька видів алгоритмів, до яких відносять лінійні, з розгалуженням, циклічні, з підпрограмами та змішані.

Таблиця 1.1 – Основні символи для побудови алгоритмів

№ зп	Символ	Призначення символу в алгоритмі
1		Термінатор – символ, що відображає початок та завершення блок-схеми алгоритму
2		Процес – символ, що дозволяє відобразити функції обробки даних будь-якого типу, тобто виконання певної операції чи декількох операцій, що призводять до відповідних змін значення, форм або розміщення заданої інформації чи визначається напрямом руху потоку, що може бути направлений за декількома напрямками
3		Розв’язок – символ відображення вирішення або функції перемикаючого типу, що містить один вхід та ряд альтернативних виходів, причому лише один з них може бути активований після обчислення встановлених умов, тобто даний символ здійснює перевірку умови
4		Дані – символ відображення даних, де носій їх не визначений, тобто введення та вивід даних на екран
5		Зумовлений процес – символ процесу, що містить одну чи декілька операцій або кроків програми, що визначені в підпрограмі чи модулі
6		Оперативний запам’ятовуючий пристрій – символ відображення даних, які зберігаються в оперативному запам’ятовуючому пристрої

Продовження таблиці 1.1 –

№ зп	Символ	Призначення символу в алгоритмі
7		Документ – символ відображення даних, що представлені на носію в будь-якій формі, що дозволяє зчитувати інформацію
8		Підготовка – символ, що зображує модифікацію однієї чи декількох команд, спрямованих впливати на наступну функцію, наприклад, ініціалізація програми
9		Запам'ятовуючі дані – символ відображення зберігаємих даних у вигляді, що дозволяють обробити інформацію, та носій їх не визначений
10		З'єднувач – символ, що відображає вихід в частині блок-схеми та вхід з іншої її частини, тим самим використовується для обриву ліній та продовження її в іншому місці в межах одного аркушу
11		Міжсторінковий з'єднувач – символ, що відображу зв'язок між інформацією на різних аркушах

Лінійні алгоритми – це алгоритми, в яких виконання кроків виконується в послідовному напрямку, тобто один за одним (рис. 1.1, а).

Розгалужені алгоритми – це алгоритми, де здійснюється перевірка певної умови, саме від якої і залежить виконання подальших кроків, і має хоча б одну базову структуру розгалуження (рис. 1.1, б)

Циклічні алгоритми – алгоритми багаторазового скінченного виконання відповідних дій. Такі алгоритми розподіляють на алгоритми з циклами з передумовою, з післяумовою та з параметрами (рис. 1.2).

Алгоритми з підпрограмою – це алгоритми, в яких застосування підпрограм дозволяє виконувати звернення до програми з різних частин основної програми.

Змішані алгоритми – це алгоритми, що містять в собі як лінійні і циклічні, так і альтернативні ділянки в структурі.

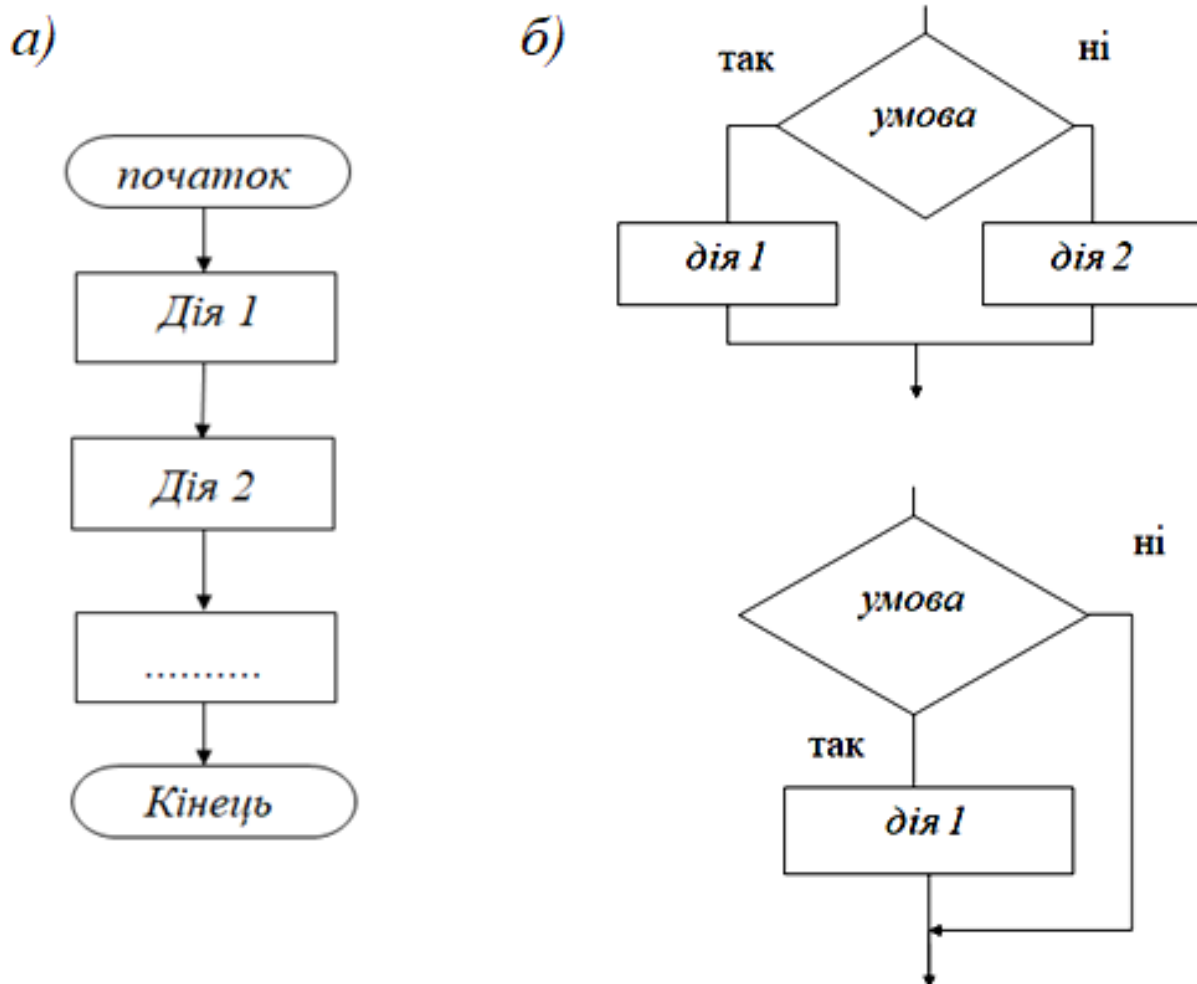


Рисунок 1.1 – Приклади блок-схем алгоритмів лінійної (а) та розгалуженої (б) структур

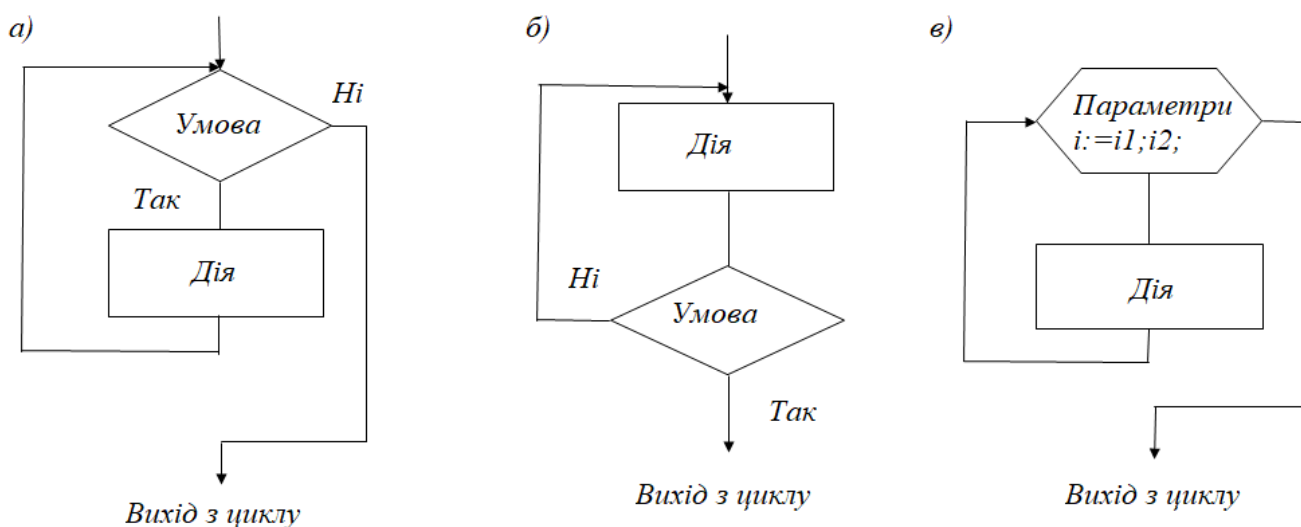


Рисунок 1.2 – Приклад алгоритмів циклічної структури: цикл з передумовою (а), цикл з післяумовою (б) та цикл з параметром (в)

Завдання до практичної роботи №1

Завдання 1. Побувати блок-схему алгоритму для визначення функції Y (у відповідності до варіанту з табл. 1.2) в заданій точці та виконати опис порядку його роботи.

Таблиця 1.2 – Вихідні дані

№ варіанту	Задана функція Y
1	$y = \begin{cases} x^{e^x} + e^2, & \text{якщо } 1 \leq x < 2 \\ \sin x^2 + \frac{x+3}{9}, & \text{якщо } -1 < x < 1 \\ \sqrt[2]{1 + \sqrt{ x-5 }}, & \text{якщо } -5 \leq x < -1 \end{cases}$
2	$y = \begin{cases} \ln^2 x^2 + \sqrt{x}, & \text{якщо } 3 < x < 6 \\ \frac{x}{3 + \frac{x^2}{5+x}}, & \text{якщо } x > 10 \\ x \ln x + \arctg x, & \text{якщо } 6 \leq x < 10 \end{cases}$

Продовження таблиці 1.2-

№ варіанту	Задана функція Y
3	$y = \begin{cases} \frac{x^3}{3 + \frac{x^3}{5}}, \text{ якщо } x \leq -1 \\ \sqrt{1 + x^2}, \text{ якщо } -1 \leq x < 0 \\ \frac{1 + x}{1 + \sqrt[3]{1 + e^{-0.2x}}}, \text{ якщо } x = 5 \text{ або } x = 10 \end{cases}$
4	$y = \begin{cases} \frac{\sqrt{1 + x }}{2 + x }, \text{ якщо } x \leq 0 \\ \frac{1 + x}{2 + \cos^3 x}, \text{ якщо } x > 1 \\ 4\sqrt{1 + x^3}, \text{ в інших випадках} \end{cases}$
5	$y = \begin{cases} \sqrt{1 + x }, \text{ якщо } x \leq 0 \\ \frac{1 + 3x}{2 + \sqrt[3]{1 + x}}, \text{ якщо } x > 1 \\ \operatorname{arctg} \frac{3x}{2}, \text{ якщо } x = 0,33 \end{cases}$
6	$y = \begin{cases} 2xe^{2x} + e^3, \text{ якщо } 0 \leq x < 3 \\ \cos x^2 + \frac{2x}{4}, \text{ якщо } -1 < x < 1 \\ \sqrt[2]{2 + \sqrt{ x - 12 }}, \text{ якщо } -12 \leq x < -1 \end{cases}$
7	$y = \begin{cases} \ln x^3 + 2x\sqrt{x}, \text{ якщо } 4 < x < 8 \\ \frac{x}{5 + \frac{x^3}{1 + 3x}}, \text{ якщо } x > 9 \\ x \cos x + \operatorname{arcsin} x, \text{ якщо } 8 \leq x < 9 \end{cases}$
8	$y = \begin{cases} \sqrt{4 - x }, \text{ якщо } -1 < x \leq 0 \\ 1 + \cos^3 x, \text{ якщо } x > 1 \\ \sqrt{1 + x}, \text{ в інших випадках} \end{cases}$

Продовження таблиці 1.2-

№ варіанту	Задана функція Y
9	$y = \begin{cases} \frac{\sqrt{1+ x }}{2+ x }, \text{ якщо } -1 < x \leq 0 \\ 2 + \sqrt[3]{1+x}, \text{ якщо } x > 1 \\ \arcsin \frac{4x}{3}, \text{ якщо } x = 0,1 \end{cases}$
10	$y = \begin{cases} \frac{x}{5 + \frac{x^3}{1+3x}}, \text{ якщо } x \leq -1 \\ \sqrt{1+x^2}, \text{ якщо } -1 \leq x < 0 \\ \frac{2+x}{x + \sqrt[4]{1+e^{-0.9x}}}, \text{ якщо } x = 4 \text{ або } x = 7 \end{cases}$
11	$y = \begin{cases} x^{e^x} + e^2, \text{ якщо } 1 \leq x < 2 \\ \cos x^2 + \frac{2x}{4}, \text{ якщо } -1 < x < 1 \\ \arcsin \frac{4x}{3}, \text{ якщо } x = 0,5 \end{cases}$
12	$y = \begin{cases} 1 + \cos^3 x, \text{ якщо } -1 < x \leq 0 \\ 2 + \sqrt[5]{1+3x}, \text{ якщо } x > 1 \\ x\sqrt{1+x}, \text{ в інших випадках} \end{cases}$
13	$y = \begin{cases} \ln x^3 - \frac{x}{3}, \text{ якщо } -1 < x < 5 \\ 5x^2 + \frac{x^3}{1+3x}, \text{ якщо } x > 5 \\ x^4 \cos x + \sin x, \text{ в інших випадках} \end{cases}$
14	$y = \begin{cases} \frac{x^3}{3 + \frac{x^3}{5}}, \text{ якщо } x \leq -1 \\ 5x^2 + \frac{x^3}{1+3x}, \text{ якщо } x > 10 \\ 1 + \sqrt[3]{1+e^{-0.2x}}, \text{ якщо } x = 4 \text{ або } x = 10 \end{cases}$

Продовження таблиці 1.2-

№ варіанту	Задана функція Y
15	$y = \begin{cases} \ln 3x^3 - \frac{2x}{3}, \text{ якщо } x \leq 0 \\ \frac{1+x}{2+\cos^3 x}, \text{ якщо } x > 1 \\ 1 + \sqrt[3]{1+e^{-0.2x}}, \text{ в інших випадках} \end{cases}$
16	$y = \begin{cases} \sqrt{1+ x }, \text{ якщо } x \leq 0 \\ \cos x^2 + \frac{2x}{4}, \text{ якщо } x > 1 \\ \arccos \frac{3x}{9}, \text{ якщо } x = 0,25 \end{cases}$
17	$y = \begin{cases} \ln^2 x^2 + \sqrt{x}, \text{ якщо } 3 < x < 6 \\ \frac{1+x}{2+\cos^3 x}, \text{ якщо } x > 6 \\ x^3 \ln x, \text{ якщо } 6 \leq x < 10 \end{cases}$
18	$y = \begin{cases} \frac{1+3x}{2+\sqrt[3]{1+x}}, \text{ якщо } x \leq -1 \\ \frac{\sqrt{1+ x }}{2+ x }, \text{ якщо } -1 \leq x < 0 \\ 1 + \frac{x}{6} \\ x^2 + \sqrt[3]{1+e^{-0.2x}}, \text{ якщо } x = 5 \text{ або } x = 10 \end{cases}$
19	$y = \begin{cases} \frac{\sqrt{1+ x }}{2+ x }, \text{ якщо } x \leq 0 \\ \frac{1+x}{2+\cos^3 x}, \text{ якщо } x > 1 \\ x^{-2}\sqrt{1+x^3}, \text{ в інших випадках} \end{cases}$
20	$y = \begin{cases} \frac{\sqrt{1+ x }}{x+1}, \text{ якщо } x \leq 0 \\ \ln x^3 - \frac{4x^2}{3}, \text{ якщо } x > 1 \\ 1 + \sqrt[5]{1+e^{-x^2}}, \text{ якщо } x = 0,33 \end{cases}$

Завдання 2. Побудувати блок-схеми алгоритму для циклів з передумовою та післяумовою щодо визначення функції Y (у відповідності до варіанту з табл. 1.3), якщо значення x буде змінюватися від X_H до X_K з відповідним кроком h .

Таблиця 1.3 – Вихідні дані

№ варіанту	Задана функція Y	Вихідні параметри				
		a	b	X_H	X_K	h
1	$y = \frac{\arctg bx}{a + \sin^2 x}$	1,2	0,75	1,35	6,5	0,05
2	$y = \frac{a \ln x^2}{b + \sqrt{\frac{x}{4}}}$	0,9	0,66	1,28	4,2	0,1
3	$y = \frac{\sin x^a}{bx^2 + a}$	0,5	1,4	4,2	8,3	0,15
4	$y = \frac{\sqrt{ax + b^2}}{bx - 1}$	0,25	0,75	6,5	9,3	0,2
5	$y = \frac{e^{ax} + b}{\arctg bx}$	0,6	0,45	4,2	8,6	0,25
6	$y = \frac{ax + b \sqrt[3]{\frac{x}{6}}}{\sin x^b}$	1,3	0,9	8,3	11,3	0,3
7	$y = \frac{x^2 \sin x^b}{ax^3 + b}$	1,9	0,12	3,1	5,9	0,35
8	$y = \frac{x^2 \sin x^b}{e^{ax} + b}$	0,95	1,1	0	4,1	0,4
9	$y = \frac{\cos(a + bx)}{ax^{e^x} + \sin^2 x}$	0,36	0,89	0,2	1,2	0,45
10	$y = \frac{ax^{e^x} + \sin^2 x}{bx^{e^x} - \cos^2 x}$	0,15	1,3	0,75	1,8	0,5

Продовження таблиці 1.3 -

№ варіанту	Задана функція Y	Вихідні параметри				
		a	b	X_H	X_K	h
11	$y = \frac{(ax + b) \ln x e^{2x}}{bx + \sqrt{\frac{ax}{2}}}$	1,25	1,38	4,2	6,5	0,55
12	$y = \frac{bx + \sqrt{\frac{ax}{2}}}{e^{ax} - bx}$	0,6	1,9	5,3	9,1	0,6
13	$y = \frac{ax e^x + \sin^2 x e^x}{b \cos^2 x}$	0,3	0,95	0,3	2,2	0,65
14	$y = \frac{(ax + b) \cos a^x}{\ln x e^{2x}}$	0,4	0,36	4,1	8,6	0,7
15	$y = \frac{x^2 \sin x^b}{\ln x e^{2x} + a}$	0,8	1,1	3,5	4,9	0,75
16	$y = \frac{\arctg bx}{ax e^{2x} - \sin e^{2x}}$	0,75	1,3	0,4	3,3	0,8
17	$y = \frac{bx + \sqrt{\frac{ax}{2}}}{\arctg ax e^{bx}}$	0,3	1,26	0,2	5,2	0,85
18	$y = \frac{bx^3 + \sqrt{\frac{ax}{4}}}{(a + bx^2) \ln x e^{2x}}$	0,9	1,5	0,75	3,9	0,9
19	$y = \frac{a + tg \frac{x}{b}}{e^{bx} - a^x}$	0,12	0,36	4,9	6,8	0,95
20	$y = \frac{ax^b + \sqrt[3]{\frac{x^a}{9}}}{\cos x^a}$	1,1	2,5	5,8	9,4	1

Зміст звіту

1. Мета та завдання до практичної роботи №1
2. Отримані блок-схеми розгалуженої та циклічної структури
3. Опис порядку роботи алгоритмів за отриманими блок-схемами
4. Відповіді на контрольні запитання
5. Висновки за отриманими результатами

Контрольні запитання

1. Дайте визначення блок-схеми.
2. Яка алгоритмічна структура називається лінійною?
3. Яка алгоритмічна структура називається структурою розгалудження?
4. Яким чином в блок-схемі позначається:
 - a. початок роботи;
 - b. введення чи виведення даних;
 - c. обчислювальний процес;
 - d. розгалудження?
5. Чи дозволено записувати в одному блоці декілька дій?
6. Яка алгоритмічна структура називається циклічною?
7. Назвіть відомі вам типи циклів.
8. В яких випадках кажуть, що відбулося зациклювання?
9. Яке значення кроку циклу не можна вживати?

Практична робота №2

ОСНОВИ ПРОГРАМУВАННЯ В ПРОГРАМНОМУ СЕРЕДОВИЩІ DEV-C++

Мета: навчитися основам роботи з інтегрованим середовищем DEV-C++ (програмна мова C) та здійснити виконання найпростішої програми

Короткі теоретичні відомості

На сьогоднішній день для забезпечення перетворень програмних кодів, що написані на мові програмування C/C++, необхідно виконати певні дії, що наведені на рис. 2.1.

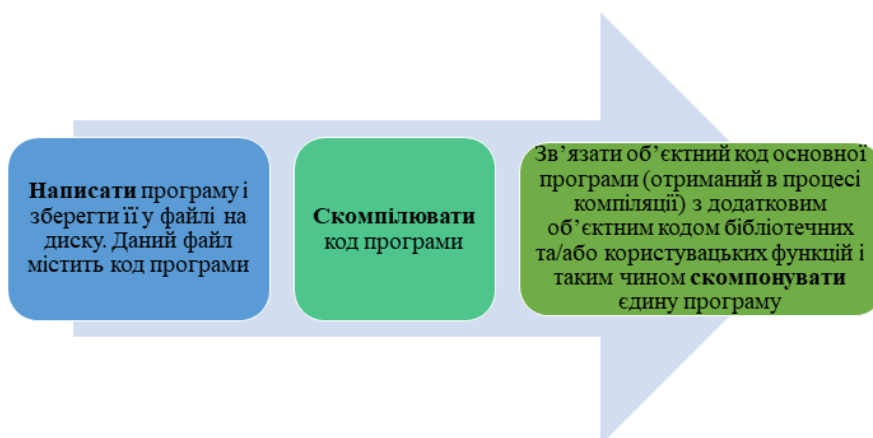


Рисунок 2.1 – Етапи створення програм

Dev-C++ - це інтегроване програмне середовище, яке забезпечує розробку різноманітних програм на мові програмування C/C++. У складі установчого пакету (дистрибутив) Dev-C++ міститься компілятор MinGW. Даний застосунок можна завантажити за посиланням <http://sourceforge.net/projects/orwelldevcpp/>.

Розробка програмних застосунків в Dev-C++ починається з його запуску IDE . Напочатку створюється новий проєкт за допомогою Файл →

Створити → Проект, а потім з'явиться діалогове вікно «Новий проєкт», що наведене на рис. 2.2. В даному вікні обирається тип проєкту, мова програмування C/C++, задається ім'я та підтверджується його створення.

Надалі формується програмний файл `main.cpp` (рис.2.3), якщо буде виникати необхідність у додаванні нових файлів до проєкту, то застосовується Файл → Створити. Права частина буде відображати сам текст файлу проєкту. Наступною дією після написання коду є компіляція Виконати → Скомпілювати (клавіша F9 або ). Якщо компіляція пройшла, тоді запускається розроблена програма за допомогою меню Виконати → Виконати чи клавішею F10 (). Також, якщо це новостворений проєкт, то в перший раз його можна відразу запустити на виконання та буде запропоновано скомпілювати проєкт за допомогою F11 або .

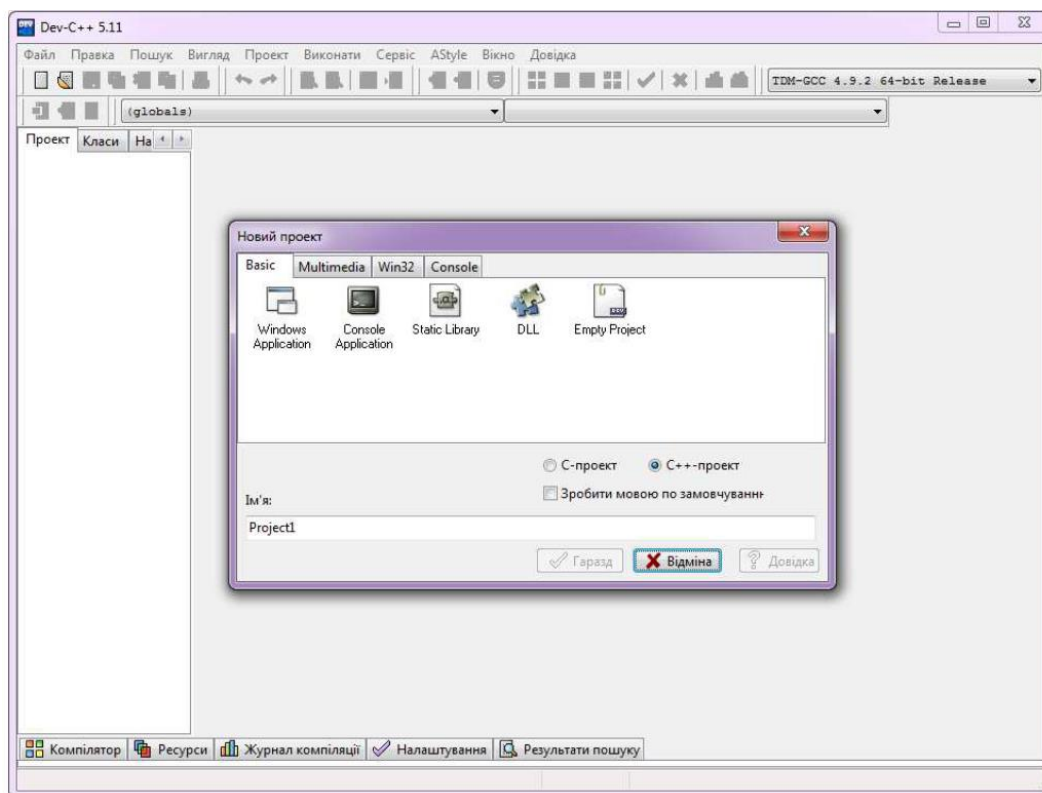


Рисунок 2.2 – Діалогове вікно «Новий проєкт»

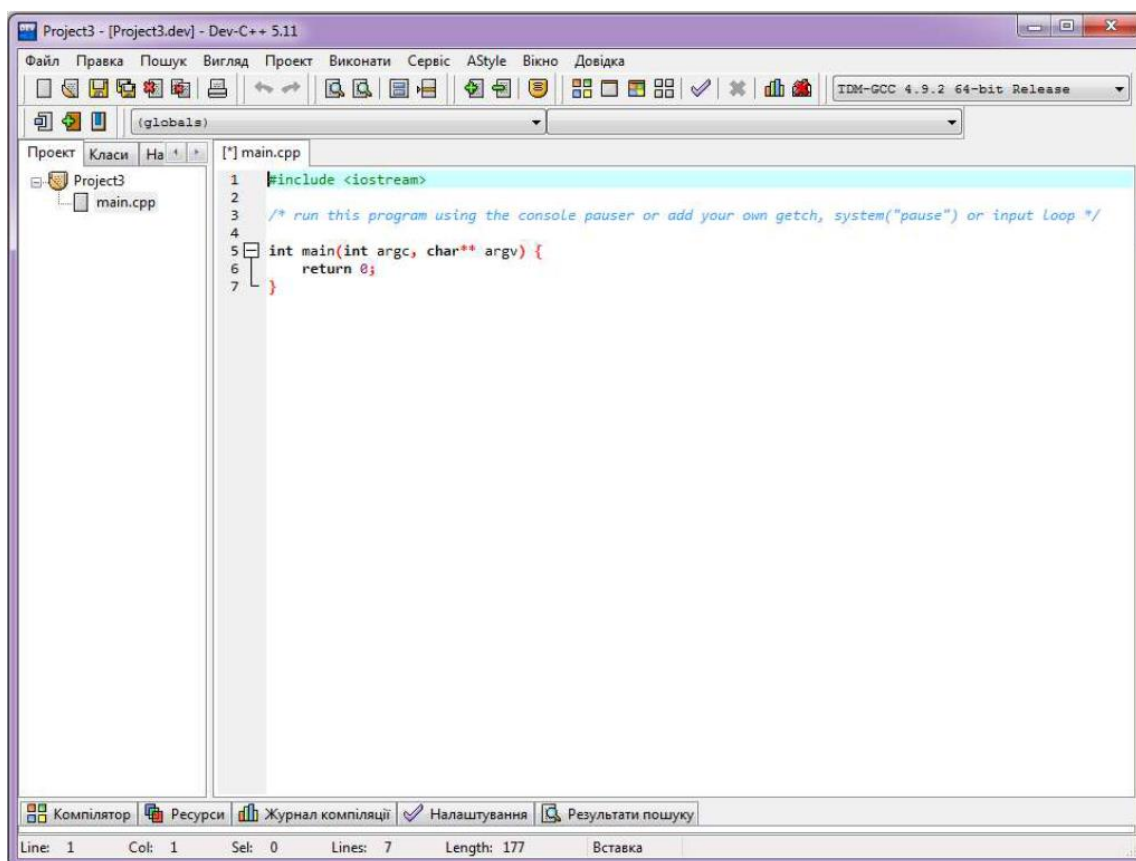


Рисунок 2.3 – Діалогове вікно розробки C++-проекту в програмному середовищі Dev-C++

У випадку виникнення помилок при розробці необхідно здійснити зневадження, тобто в перший раз необхідно за допомогою меню Виконати → Налаштування здійснити підтвердження на його дозвіл. Для забезпечення зневадження встановлюються точки зупину у відповідних рядках клавішею F4 з натисканням лівої кнопки миші на номері відповідного рядка програмного коду зліва чи скориставшись Виконати → Встановити/зняти точку зупину. Визначивши точки зупину, необхідно на панелі Налаштування натиснути кнопку Налаштування та виконувати переміщення за лістингом програми поміж точками зупину (рис. 2.4) за допомогою кнопок, що наведені рис. 2.5.

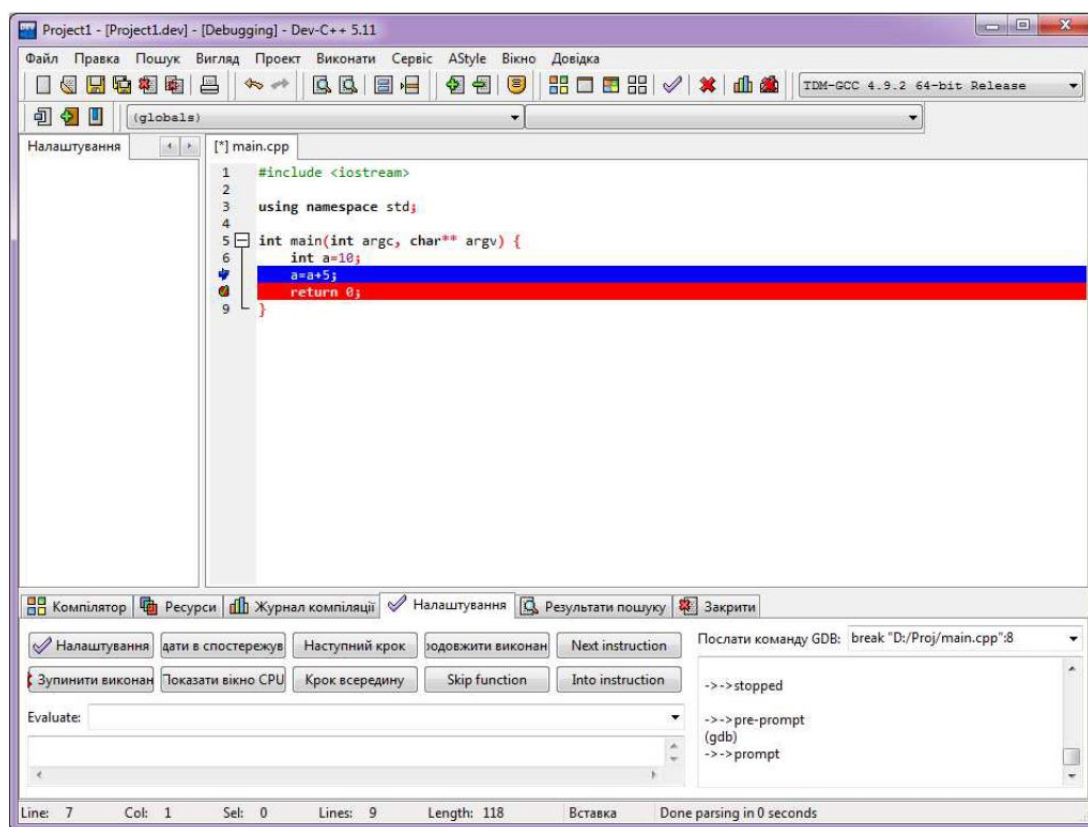


Рисунок 2.4 – Виконання «Зневадження проекту» в програмному середовищі Dev-C++

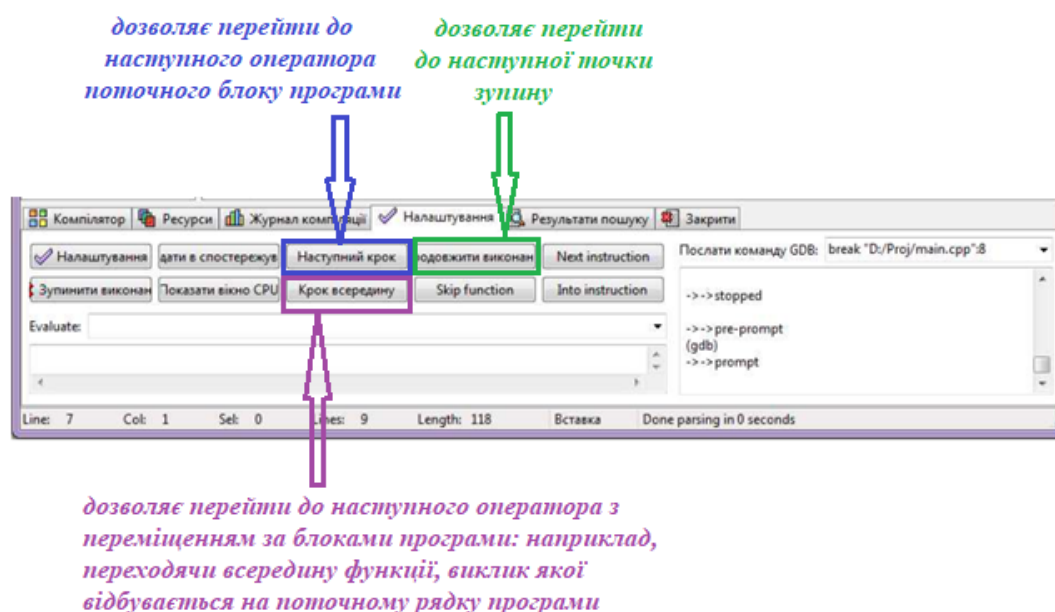


Рисунок 2.5 – Основні кнопки переміщення за лістингом програми поміж точками зупину на панелі Налаштування

Коли процес зневадження програми буде зупинено у певній (відповідній) точці зупини, тоді при наведенні курсору на змінну буде відтворюватися її поточне значення (рис. 2.6).

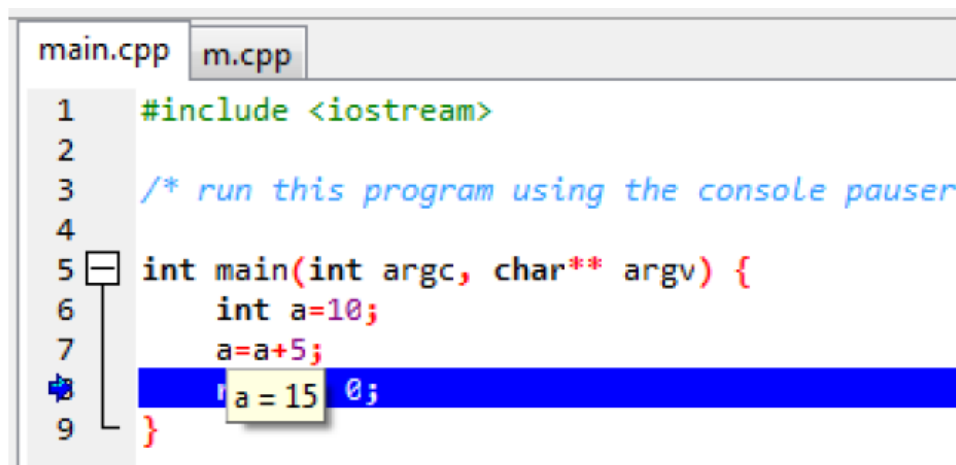


Рисунок 2.6 – Результат відображення значення змінної після виконання зневадження

Завдання до практичної роботи №2

1. Ознайомитися з програмним середовищем Dev-C++.
2. Створити новий проєкт з назвою *pr2_номер_завдання_Прізвище_Ім'я_група*
3. Розробити програму, що дозволяє виводити на консоль у різних рядках наступні елементи: назва університету, факультету, групи, прізвище, ім'я та по батькові.
4. Зробити перевірку працездатності коду програми, що виконує обчислення довжини кола (рис. 2.7).
5. Виконати модифікацію лістингу (рис. 2.7) з огляду на обчислення довжини кола та його площі для будь яких радіусів.

```

main.cpp  +
1  #include<stdio.h>
2  /*Обчислення довжини кола */
3  main ()
4  {
5  int radius;
6  float length;
7  printf("Введення значення радіусу :\n");
8  scanf ("%d", &radius);
9  length = 3.1415*2*radius;
10 printf("Радіус-%d\n довжина - %f\n", radius,
11 length);
12 }

```

Рисунок 2.7 – Лістинг програми обчислення довжини кола

6. Виконати програму з лістингу (рис. 2.8) та перевірити дію інкрементних операторів присвоєння.

```

main.cpp  +
1  #include <stdio.h>
2  #include <conio.h>
3  main ( )
4  {
5  int a = 5, b = 6, sum;
6  sum = a + b; printf ("a=%d b=%d sum=%d\n", a, b, sum);
7  sum =a + + +b;printf("a=%d b=%d sum=%d\n", a, b, sum);
8  sum =++a + b; printf (("a=%d b=%d sum=%d\n" ,a,b, sum);
9  sum = - -a + b; printf(("a=%d b=%d sum=%d\n",a,b, sum);
10 sum = a- - +b; printf(("a=%d b=%d sum=%d\n",a, b, sum);
11 sum = a + b;printf (("a=%d b=%d sum=%d\n" , a, b, sum);
12 getch();
13 }

```

Рисунок 2.8 – Лістинг програми до завдання №8

Зміст звіту

1. Мета та завдання до практичної роботи №2
2. Лістинги розроблених програм із коментарями
3. Результати розрахунків за отриманими програмами
4. Відповіді на контрольні запитання
5. Висновки за отриманими результатами

Контрольні запитання

1. Які компілятори мови C/C++ Вам відомі?
2. Де зберігається проект?
3. Де зберігається вихідний файл створеного проекту?
4. Яке призначення заголовних файлів `stdio.h`, `conio.h`?
5. Назвіть функції виведення інформації на консоль.
6. Назвіть функцію введення інформації. Яку особливість її запису ви знаєте?
7. Як здійснюється табуляція рядка на консолі і на скільки позицій виконується відступ від лівого краю?
8. Який символ управління дозволяє виводити інформацію з нового рядка?
9. Якими комбінаціями символів позначають коментарі?
10. Назвіть формати для введення цілих чисел, одиночного символу, рядка символів, чисел з плаваючою комою.

Практична робота №3

ОРГАНІЗАЦІЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ

Мета: отримати практичні навички при організації розгалужених процесів мовою програмування C.

Короткі теоретичні відомості

Розгалуження в програмних застосунках виконується за допомогою оператора умов **if**, причому він використовується у двох різних формах (табл. 3.1).

Таблиця 3.1 – Форми оператора умов **if**

Форма	Формальний опис		Виконання
повна	if (вираз) оператор 1; else оператор 2;	if – якщо else – інакше	В даній формі виконується правило, де виконання оператора if починається з обчислення виразу, тому якщо вираз у дужках є істинним (не дорівнює 0), тоді виконується оператор 1, а в протилежному випадку – оператор 2.
скорочена	if (вираз) оператор 1;	if – якщо	В скороченій формі якщо вираз у дужках є істинним (не дорівнює 0), тоді виконується оператор 1, тоді як в протилежному випадку керування передається наступному оператору.

Перевірка істинності виразу виконується за допомогою операцій відношень (табл. 3.2).

Таблиця 3.2 – Операції відношень для перевірки істинності виразів

№	Операція відношення	Умовне позначення	Примітка
1	Рівність	$=$	Якщо операнди дорівнюють, то вираз істинний, в противному випадку - неправда
2	Нерівність	$\neq$	Якщо операнди не дорівнюють, то вираз істинний, в противному випадку - неправда
3	Менше	$<$	Якщо лівий операнд менше за правий, то вираз істинний, в противному випадку - неправда
4	Більше	$>$	Якщо лівий операнд більше за правий, то вираз істинний, в противному випадку - неправда
5	Менше або рівність	$\leq$	Якщо лівий операнд менше або дорівнює правому, то вираз істинний, в противному випадку - неправда
6	Більше або рівність	$\geq$	Якщо лівий операнд більше або дорівнює правому, то вираз істинний, в противному випадку - неправда

На прикладі 3.1 представлено, що як і на місці оператора 1, так і оператора 2 можна застосувати складну структуру.

Приклад 3.1

```

if (i < j) i++;
else
{
    j = i-3; i++;
}

```

Якщо за умовою ставляться два і більше операторів, то вони укладаються у фігурні дужки. Допускається використання вкладених операторів **if**. Оператор **if** може бути включений у конструкцію **if** або в конструкцію **else** іншого оператора **if**. Щоб зробити програму більш читабельною, рекомендується групувати оператори та конструкції у вкладених операторах **if**, використовуючи фігурні дужки та відступи. Якщо ж фігурні дужки опущені, то компілятор пов'язує кожне ключове слово **else** з найближчим до нього **if**, для якого немає **else**.

Приклад 3.2

```

int main ( )
{
    int t=2, b=7, r=3;
    if (t > b)
    {
        if (b < r) r=b;
    }
    else r = t;
    return 0;
}

```

У результаті виконання цієї програми **r** стане рівним 2.

Часто вирази в операторі **if** бувають складовими, тобто перевіряється дві або більше умов. У цьому випадку їх об'єднують у вигляді зв'язаного ланцюжка умов за допомогою логічних операторів **І (&&)** та **АБО (||)**.

Приклад 3.3

```
int main( )
{
    int a, b, c;
    if ( a == x )
        b = a;
        if ( a != x  && a ==c)
            b=0;
            if ( a != x  || a!=c)
                b=1;
    return 0;
}
```

Завдання до практичної роботи №3

Скласти алгоритм та його програмну реалізацію для порівняння двох математичних виразів із таблиці 3.3. Вважати, що вирази рівні (або не рівні) якщо їхня різниця не перевищує 0,001. Параметри тригонометричних функцій ввести в градусах або радіанах. Для роботи з тригонометричними функціями підключіть директиву **#include <math.h>**.

Таблиця 3.3 – Вихідні вирази для виконання ПР3

Варіанту	Вираз 1	Вираз 2
1	$F1 = 2\sin^2(3\pi - 2\beta) \cdot \cos^2(5\pi + 2\beta)$	$F2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\beta\right)$
2	$F1 = \cos \beta + \sin \beta + \cos 3\beta + \sin 3\beta$	$F2 = 2\sqrt{2} \cos \beta \cdot \sin\left(\frac{\pi}{4} + 2\beta\right)$

Продовження таблиці 3.3 –

Варіанту	Вираз 1	Вираз 2
3	$F1 = \frac{\sin 2\beta + \sin 5\beta - \sin 2\beta}{\cos \beta + 1 - 2\sin^2 2\beta}$	$F2 = 2 \sin \beta$
4	$F1 = \frac{\sin 2\beta + \sin 5\beta - \sin 3\beta}{\cos \beta - \cos 3\beta + \cos 5\beta}$	$F2 = \operatorname{tg} 3\beta$
5	$F1 = 1 - \frac{1}{4} \sin^2 2\beta + \cos 2\beta$	$F2 = \cos^2 \beta + \cos^4 \beta$
6	$F1 = \cos \beta + \cos 2\beta + \cos 6\beta + \cos 7\beta$	$F2 = 4 \cos \frac{\beta}{2} \cdot \cos \frac{5}{2} \beta \cdot \cos 4\beta$
7	$F1 = \cos^2 \left(\frac{3}{8} \pi - \frac{\beta}{2} \right) - \cos^2 \left(\frac{11}{8} \pi - \frac{\beta}{4} \right)$	$F2 = \frac{\sqrt{2}}{2} \sin \frac{\beta}{2}$
8	$F1 = \cos^4 \beta + \sin^2 \beta + \frac{1}{4} \sin^2 2\beta - 1$	$F2 = \sin(\alpha + \beta) \cdot \sin(\alpha - \beta)$
9	$F1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2$	$F2 = -4 \sin^2 \left(\frac{\alpha - \beta}{2} \right) \cdot \cos(\alpha + \beta)$
10	$F1 = \frac{\sin \left(3\beta + \frac{\pi}{2} \right)}{1 - \sin(3\beta - \pi)}$	$F2 = \operatorname{ctg} \left(\frac{5}{4} \pi - \frac{3}{2} \beta \right)$
11	$F1 = \frac{1 - 2\sin^2 \beta}{1 - \sin(3\beta - \pi)}$	$F2 = \frac{1 - \operatorname{tg} \beta}{1 + \operatorname{tg} \beta}$
12	$F1 = \frac{\sin 4\beta}{1 + \cos 4\beta} \cdot \frac{\cos 2\beta}{1 + \cos 2\beta}$	$F2 = \operatorname{ctg} \left(\frac{3}{2} \pi - \beta \right)$
13	$F1 = \frac{\sin \beta + \cos(2\beta - \alpha)}{\cos \beta - \sin(2\beta - \alpha)}$	$F2 = \frac{1 + \sin 2\beta}{\cos 2\beta}$
14	$F1 = \frac{\sin \beta + \cos \beta}{\cos \beta - \sin \beta}$	$F2 = \operatorname{tg} 2\beta + \sin 2\beta$
15	$F1 = \frac{\sqrt{2b + 2\sqrt{bb} - 4}}{bb - 4 + b + 2}, bb = b^2$	$F2 = (\sqrt{b + 2})^2$

Продовження таблиці 3.3 –

Варіанту	Вираз 1	Вираз 2
16	$F1 = \frac{x^2 + 2x - 3 + (x + 1)\sqrt{x^2 - 9}}{x^2 - 2x - 3 + (x - 1)\sqrt{x^2 - 9}}$	$F2 = \sqrt{\frac{x + 3}{x - 3}}$
17	$F1 = \frac{\sqrt{(3x + 2)^2 - 24x}}{3\sqrt{x} - \frac{2}{\sqrt{x}}}$	$F2 = -\sqrt{x}$
18	$F1 = \left(\frac{a + 2}{\sqrt{2a}} - \frac{a}{\sqrt{2a} + 2} + \frac{2}{a - \sqrt{2a}} \right)$	$F2 = \frac{1}{\sqrt{a} + \sqrt{2}}$
19	$F1 = \left(\frac{a + 2}{\sqrt{2a}} + \frac{a}{\sqrt{2a} + 2} \right)$	$F2 = \frac{1}{\sqrt{a}} + \sqrt{2}$
20	$F1 = \frac{\sin\left(3\beta + \frac{\pi}{2}\right) + \cos\left(3\beta + \frac{\pi}{2}\right)}{\sin \beta + \cos \beta}$	$F2 = \frac{1 - \sin \beta}{1 + \operatorname{tg} \beta}$

Зміст звіту

1. Мета та завдання до практичної роботи №3
2. Алгоритм обчислення заданого викладачем варіанта
3. Таблицю змінних і трасувальну таблицю
4. Лістинг програми із коментарями та результати за отриманими програмами
5. Відповіді на контрольні запитання
6. Висновки за отриманими результатами

Контрольні запитання

1. Для якої мети використовуються розгалужені процеси?
2. Назвіть основні оператори розгалужених процесів
3. У яких випадках використовується конструкція if...else?

Практична робота №4

УМОВНІ ОПЕРАТОРИ ТА ОПЕРАТОРИ ВИБОРУ

Мета: Отримати практичні навички керування процесів виконання програм за умов виконання послідовності операторів; визначеної послідовності операторів до тих пір, доти умова істинна; перевірки істинності умови для вибору між різними, можливими способами дії.

Короткі теоретичні відомості

При складанні програми розв'язку квадратного рівняння обов'язковою умовою є використання елементів:

- файл заголовка ***math.h***, який дозволяє розрахувати квадратний корінь із дискримінанта (***sqrt(D)***);
- конструкцію виду: ***if (вираз) - оператор***, що використовується, якщо вираз істинний.

Приклад 4.1

```
// мається на увазі, що комплексних коренів немає  
if(D<0)  
printf("Розв'язку немає \n");
```

За необхідності в комбінації з ***if*** можна використовувати ключове слово ***else***, що надає можливість виконати альтернативний оператор, або блок операторів, якщо умова неістинна.

Приклад 4.2

```

if (D<0)
printf ("Розв'язку не має \n")
else
printf ("Розв'язок є \n");

```

Оператори **if** та **else** можуть бути вбудованими.

Якщо необхідно розробити діалогову програму, що дає змогу отримувати розв'язки квадратного рівняння за різними значеннями коефіцієнтів a, b, c , а також виходу з програми за запитом, тоді використовують конструкцію **while (вираз) - оператор**. Ключове слово **while** дозволяє виконувати оператор або блок до тих пір, поки умова не перестане бути істинною. Оператор або блок, що пов'язані з **while**, не буде виконуватися у випадку, коли вираз напочатку буде помилковим.

Приклад 4.3

```

char vych='d';
:
:
while(vych= 'd')
{
:
: // група операторів
printf('Продовжувати розв'язок? (d/n)\n');
cscanf("%c",&vych);
}

```

У цьому фрагменті програми блок команд виконуватиметься до тих пір, поки символьній змінній не буде присвоєно значення 'd'.

Коли необхідно отримати певний розв'язок в залежності від значень коефіцієнтів можна застосовувати в програмному застосунку наступну конструкцію:

```
switch (вираз_1)
{
case константа 1: оператор або група операторів блоку 1;
break;
:
:
case константа 2: оператор або група операторів блоку 2;
break;
default: оператор або група операторів;
}
```

При виконанні оператора **switch** спочатку обчислюється значення *виразу_1*, що стоїть у дужках даного оператора. Тип значення має бути одним із цілих: **char**, **int**, **unsigned int**, **long int** і **long unsigned**. Обчислене значення порівнюється зі значеннями констант операторів **case**.

При збігу значення *виразу_1* з і-ю константою виконується оператор або група операторів і-го блоку. Потім керування передається на наступний оператор (після **switch**), якщо в і-й гілці присутній оператор **break**.

Якщо значення *виразу_1* не збігається з жодною з констант, виконується оператор або група операторів, позначених **default**. За її відсутності виконується наступний після **switch** оператор.

Приклад 4.4

```

int var;
:
:
if (A1==0)
var=1;
else
var=2;
:
:
switch(var)
{
case 1: {
printf ("Розв'язок рівняння \n"); ...
x1 = .....;
x2 = .....;
break;
}
case 2: printf ("Розв'язок рівняння \n");
x1 = .....;
:
:
break;
default: puts (" Помилка \n");
}

```

Якщо значення $D < 0$ (корені комплексні), то передбачте зміну знаку D перед обчисленням квадратного кореня, інакше буде зафіксовано помилку.

Завдання до практичної роботи №4

1. Скласти програму розв'язку квадратного рівняння:

$$ax^2 + bx + c = 0 \quad (4.1)$$

з повним аналізом можливих розв'язків на основі конструкцій **if** і **if-else** за наступних умов:

$$D < 0, D = 0, D > 0 \quad (4.2)$$

D — дискрименат.

2. Розробити діалогову програму, що дає змогу отримувати розв'язки квадратного рівняння (4.1) за різних значень коефіцієнтів a, b, c , а також виходу з програми за запитом, використовуючи конструкції **while** або **do...while** (за вибором).

3. Розробити діалогову програму, що дає змогу залежно від значень коефіцієнтів отримувати той чи інший розв'язок, використовуючи оператор вибору **switch()**:

- а) якщо $a=0$; b і c не дорівнюють нулю;
- б) якщо $b=0$; a і c не дорівнюють нулю;
- в) якщо $c=0$; b і a не дорівнюють нулю;
- г) інші можливі поєднання коефіцієнтів a, b, c .

Програма має обчислювати як дійсні, так і комплексні корені.

4. Вивести на екран відомості про автора, вихідні значення коефіцієнтів, значення дискримінанта і результати розв'язання.

5. Розробити алгоритм і програму обчислення функції F на інтервалі від $X_{поч}$ до $X_{кін}$ з кроком dx . Результати обчислень вивести у вигляді таблиці значень функції. Варіанти завдань наведено в табл. 4.1.

Таблиця 4.1 – Вихідні дані до завдання 5

Варіант	Функції	Умова
1	$F = \begin{cases} ax^2 + b, & \text{при } x < 0, b \neq 0 \\ \frac{x-a}{x-c}, & \text{при } x > 0, b = 0, c \neq 0 \\ \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
2	$F = \begin{cases} \frac{1}{ax} - b, & \text{при } x + 5 < 0, c = 0 \\ \frac{x-a}{x-c}, & \text{при } x + 5 > 0, c \neq 0 \\ \frac{10x}{c-4}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
3	$F = \begin{cases} ax^2 + bx + c, & \text{при } a < 0, c \neq 0 \\ \frac{-a}{x-c}, & \text{при } a > 0, c = 0, x \neq c \\ \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
4	$F = \begin{cases} -ax - c, & \text{при } x < 0, a \neq 0 \\ \frac{x-a}{-c}, & \text{при } c > 0, a = 0 \\ \frac{bx}{c-a}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
5	$F = \begin{cases} a - \frac{x}{10+b}, & \text{при } x < 0, b \neq 0 \\ \frac{x-a}{x-c}, & \text{при } x > 0, b = 0 \\ 3x + \frac{2}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
6	$F = \begin{cases} ax^2 + b^2x, & \text{при } x < 0, b \neq 0 \\ \frac{x+a}{x+c}, & \text{при } c > 0, a = 0 \\ \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури

Продовження таблиці 4.1 -

Варіант	Функції	Умова
7	$F = \begin{cases} -ax^2 - b, & \text{при } a < 0, b \neq 0 \\ \frac{x-a}{x-c}, & \text{при } x > 5, c \neq 0 \\ 1 - \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
8	$F = \begin{cases} ax^2 + c, & \text{при } c < 0, b \neq 0 \\ \frac{x-a}{xc}, & \text{при } c > 0, a = 0 \\ \frac{x-c}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
9	$F = \begin{cases} ax^2 + b^2x, & \text{при } a < 0, x \neq 0 \\ x - \frac{a}{x-c}, & \text{при } a > 0, c \neq x \\ 1 + \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
10	$F = \begin{cases} ax^2 - bx, & \text{при } x < 3, b \neq 0 \\ \frac{x-a}{x-c}, & \text{при } x > 3, a = 0 \\ \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
11	$F = \begin{cases} ax^2 + \frac{b}{c}, & \text{при } x < 1, c \neq 0 \\ \frac{x-a}{(x-c)^2}, & \text{при } x > 1, c = 0 \\ \frac{x^2}{c^2}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
12	$F = \begin{cases} ax^2 + b^2 + c, & \text{при } x < 0,5, b \neq 0 \\ \frac{x-a}{x-c}, & \text{при } x > 0,6, b + c = 0 \\ \frac{x}{c} + \frac{x}{a}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури

Продовження таблиці 4.1 -

Варіант	Функції	Умова
13	$F = \begin{cases} ax^2 + b, & \text{при } x - 1 < 0, b \neq 0 \\ \frac{x - a}{x}, & \text{при } x - 1 > 0, b = 0 \\ \frac{x}{c - x}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
14	$F = \begin{cases} ax^2 + b^2, & \text{при } x + c < 0, a \neq 0 \\ \frac{x - a}{x - c}, & \text{при } x + c > 0, a = 0 \\ 1 - \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
15	$F = \begin{cases} a(x + c)^2 - b, & \text{при } x = 0, b \neq 0 \\ \frac{x - a}{c}, & \text{при } x = 5, c \neq 0 \\ a + \frac{x}{x + c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
16	$F = \begin{cases} ax^2 + b, & \text{при } x < 0, b \neq 0 \\ \frac{x}{x - c} + 10, & \text{при } x > 0, b = 0 \\ -\frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
17	$F = \begin{cases} ax^2 - cx + b, & \text{при } x + 5 < 0 \\ \frac{x - a}{x - c}, & \text{при } x + 5 > 0, c \neq x \\ -\frac{x}{a - c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
18	$F = \begin{cases} ax^2 + bx^2, & \text{при } x < 0, b \neq 0 \\ \frac{x - a}{x - c}, & \text{при } x > 0, c \neq x \\ \frac{x + 5}{c(x - 10)}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури

Продовження таблиці 4.1 -

Варіант	Функції	Умова
19	$F = \begin{cases} a(x+8)^2 - b, & \text{при } x < 5, b \neq 0 \\ \frac{x - cb}{c + 5}, & \text{при } x > 5, b = 0 \\ \frac{x}{c}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури
20	$F = \begin{cases} -\frac{2x - c}{cx - a}, & \text{при } x < 0, c \neq 0 \\ \frac{x - a}{x - c}, & \text{при } x > 0, x - c \neq 0 \\ -\frac{x}{c} - \frac{c}{2x}, & \text{в інших випадках} \end{cases}$	a, b, c – дійсні числа. Значення $a, b, c, X_{поч}, X_{кін}, dx$ вводяться з клавіатури

Зміст звіту

1. Мета та завдання до практичної роботи №4
2. Структурні схеми алгоритмів розв'язання задачі
3. Лістинги програм із коментарями та отримані результати
4. Відповіді на контрольні запитання
5. Висновки за отриманими результатами

Контрольні запитання

1. Які керуючі засоби виконання програм існують у мові Сі? Поясніть їхній синтаксис.
2. Поясніть призначення всіх функцій, використаних у програмах.
3. Які операції використовуються в керуючих засобах виконання програм.
4. Поясніть відмінності між циклами while і do ... while.
5. Для яких цілей використовується оператор switch ?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. ДСТУ ISO 5807:2016 (ISO 5807:1985, IDT). Оброблення інформації. Символи та угоди щодо документації стосовно даних, програм та системних блок-схем, схем мережевих програм та схем системних ресурсів : чинний від 2016-10-10. – Київ, 2016. – (Національний стандарт України).
2. Вступ до програмування мовою C++. Організація обчислень : навч. посіб./ Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль та ін. – К. : Вид.-поліграф. центр "Київський ун-т", 2012. – 175 с.
3. Бублик, В.В. Об'єктно-орієнтоване програмування : підручник / В.В. Бублик. – К. : ІТкнига, 2015. – 624 с.
4. Жуковський, С.С. Об'єктно-орієнтоване програмування мовою C++ : навч.-метод. посіб. для студ. напряму 6.040302 Інформатика / С.С. Жуковський, Т.А. Вакалюк. – Житомир : Вид-во ЖДУ, 2016. – 100 с.
5. Карпенко, Н.В. Розробка програм на мові С у сучасних середовищах : навч.-метод. посіб. / Н.В. Карпенко – Д. : ЛПРА, 2016. – 144 с.
6. Методичні вказівки до виконання самостійної роботи студентів та розрахунково-графічних завдань з дисципліни “Алгоритмізація та програмування” для студентів спеціальності 122 “Комп’ютерні науки” (всіх форм навчання) / В. М. Льовкін. – Запоріжжя : ЗНТУ, 2017. – 54 с.
7. Яворський, Н. Б. Лабораторний практикум з дисципліни “Алгоритмізація та програмування”: навч. посіб. / Н. Б. Яворський, У. Б. Марікуца, М. І. Андрійчук та ін. – Львів : Вид-во Львівської політех., 2018. – 191 с.
8. Трофименко, О.Г. C++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логіновата ін. – 2-ге вид. перероб. і допов. – Одеса : Фенікс, 2019. – 477 с.

9. Обчислювальна техніка та програмування: Алгоритми та їх реалізація. Комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 141 Електроенергетика, електротехніка та електромеханіка та 184 Гірництво / КПІ ім. Ігоря Сікорського ; уклад. : Д. В. Філянін, В. П. Опришко, В. О. Броницький та ін. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 95 с.

10. Основи алгоритмізації та програмування: навч. посібник / Л.М. Бандоріна, Т.О. Климкович, К.О. Удачина. – Дніпро : УДУНТ, 2022. – 158 с.

11. DEV C++ Blog [Електронний ресурс]. – Режим доступу: <http://orwelldevcpp.blogspot.com/>

12.wxDev-C++ [Електронний ресурс]. – Режим доступу: <http://wxdsgn.sourceforge.net/>

13.Programming with wxDevC++ [Електронний ресурс]. – Режим доступу: <http://wxdevcppbook.sourceforge.net/>