МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

Кафедра: “Систем штучного інтелекту та аналізу даних”

МЕТОДИЧНІ ВКАЗІВКИ

**для виконання лабораторних робіт
з дисципліни**

ПРОГРАМУВАННЯ

Частина 1

для студентів спеціальностей

**F3 Комп’ютерні науки,
F4 Системний аналіз та наука про дані,
F6 Інформаційні системи і технології,
F7 Комп’ютерна інженерія.**

ТЕРНОПІЛЬ 2026

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

УДК 681,5
М 54

Методичні вказівки до виконання лабораторних робіт з курсу "Програмування". Частина 1. Для студентів спеціальностей: F3 "Комп'ютерні науки", F4 "Системний аналіз та наука про дані", F6 "Інформаційні системи і технології ", F7 "Комп'ютерна інженерія". Упоряд.: Ю. Гладьо, Г. Семенишин, С. Гладьо, Н.Гащин – Тернопіль: видавництво ТНТУ, 2026. – 44с

Методичні вказівки розроблені у відповідності з навчальними планами F3 "Комп'ютерні науки", F4 "Системний аналіз та наука про дані", F6 "Інформаційні системи і технології ", F7 "Комп'ютерна інженерія".

Укладачі: к.т.н., доцент Гладьо Ю.Б.,
старший викладач Семенишин Г.М.,
к.т.н. старший викладач Гладьо С.В.
к.т.н., доцент Гащин Н.Б.

Рецензент: к.т.н., доцент Дуда О.М.

Затверджено на засіданні кафедри систем штучного інтелекту та аналізу даних

Протокол № 3 від *15 березня 2026 р.*

Схвалено та рекомендовано до друку методичною комісією факультету комп'ютерно-інформаційних систем і програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя.

Протокол № 4 від *9 квітня 2026 р.*

Лабораторна робота №1
(2 год.)

Тема: Складання блок-схем алгоритму найпростіших обчислювальних процесів.

Мета роботи: Освоїти методи розробки та складання блок-схем алгоритму розгалужених та циклічних обчислювальних процесів.

За час виконання лабораторної роботи студент повинен освоїти:

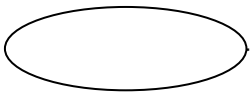
- Вид та призначення блоків схем алгоритму.
- Основні прийоми розробки схем алгоритму, реалізації розгалужених та циклічних обчислювальних процесів.
- Організація вводу, обробки та виводу при обробці інформації у виді одно і двовимірних масивів.
- Методи тестування блок-схем алгоритмів.

Порядок виконання лабораторної роботи:

- Проаналізувати задачу та вибрати метод побудови блок-схеми алгоритму.
- Виділити в проекті блок-схеми алгоритму частини вводу інформації, її обробки та виводу результатів.
- Оформити готові блок-схеми алгоритму, супроводжуючи їх необхідними коментарями.
- Провести тестування створених блок-схем алгоритму.
- Оформити звіт про виконану роботу

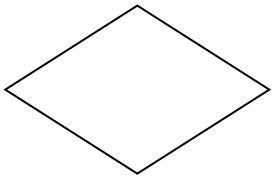
Теоретичні відомості

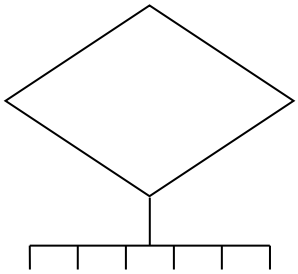
Основні блоки схем алгоритму:

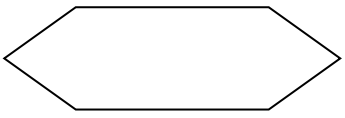

– блок початку або кінця;

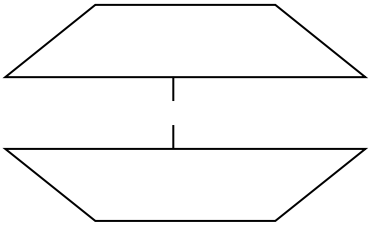

– ввід або вивід;

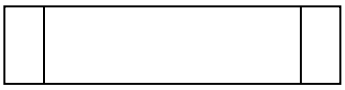

– блок обчислення (процес);

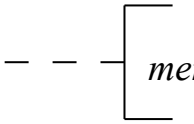

– розгалуження;

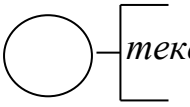

– вибір;


– модифікація (цикл з кроком);


– цикл по умові;

 – наперед визначений процес (перехід до підпрограм);

 – коментар;

 – односторінковий з'єднувач;  міжсторінковий з'єднувач;

Блоки схем алгоритму можуть нумеруватися чи позначатися. Номер (мітка) записується зліва над блоком. Обов'язково повинні позначатися блоки, з'єднування між якими здійснюється при допомозі з'єднувачів.

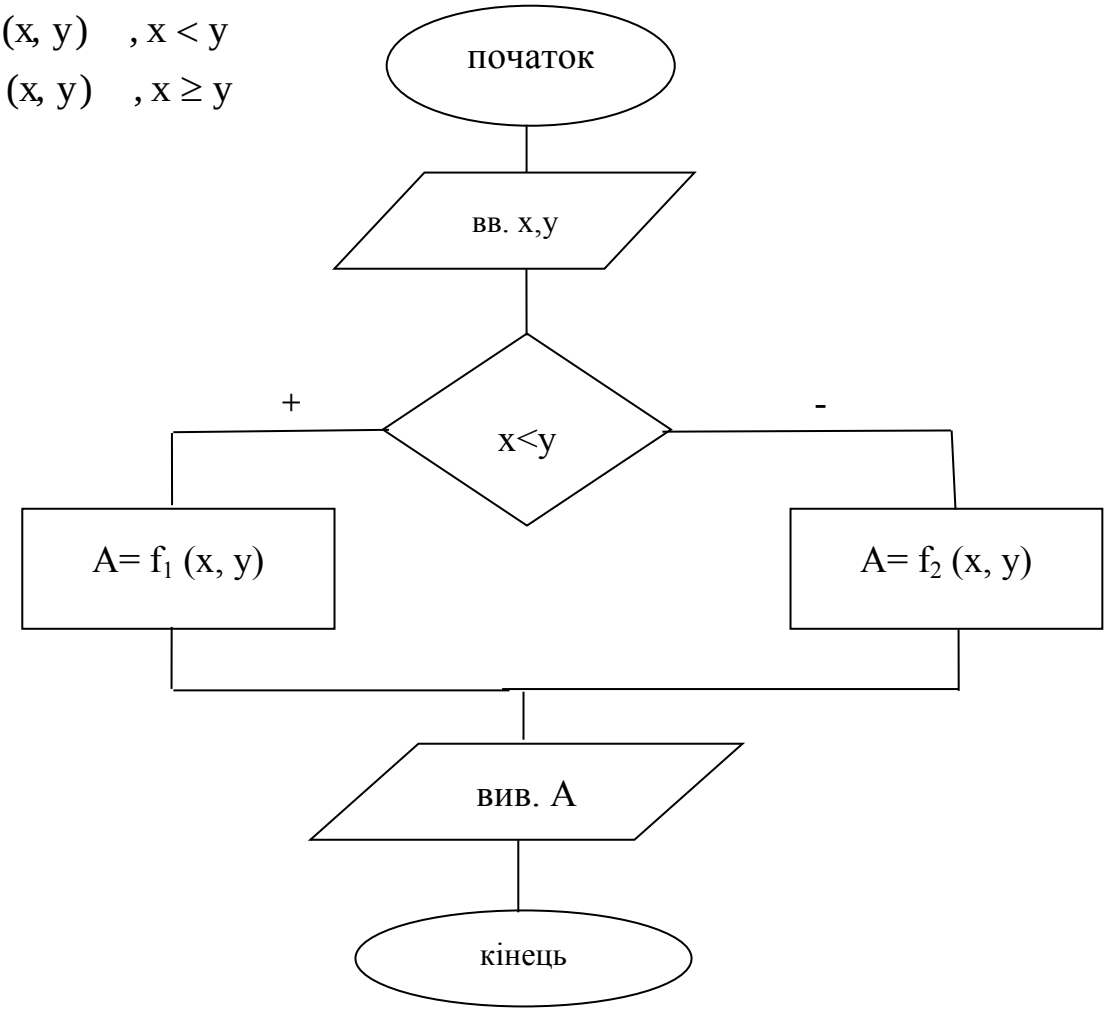
.

Розгалужений обчислювальний процес:

<u>Для правильної побудови алгоритму розгалуженого обчислювального процесу необхідно:</u>

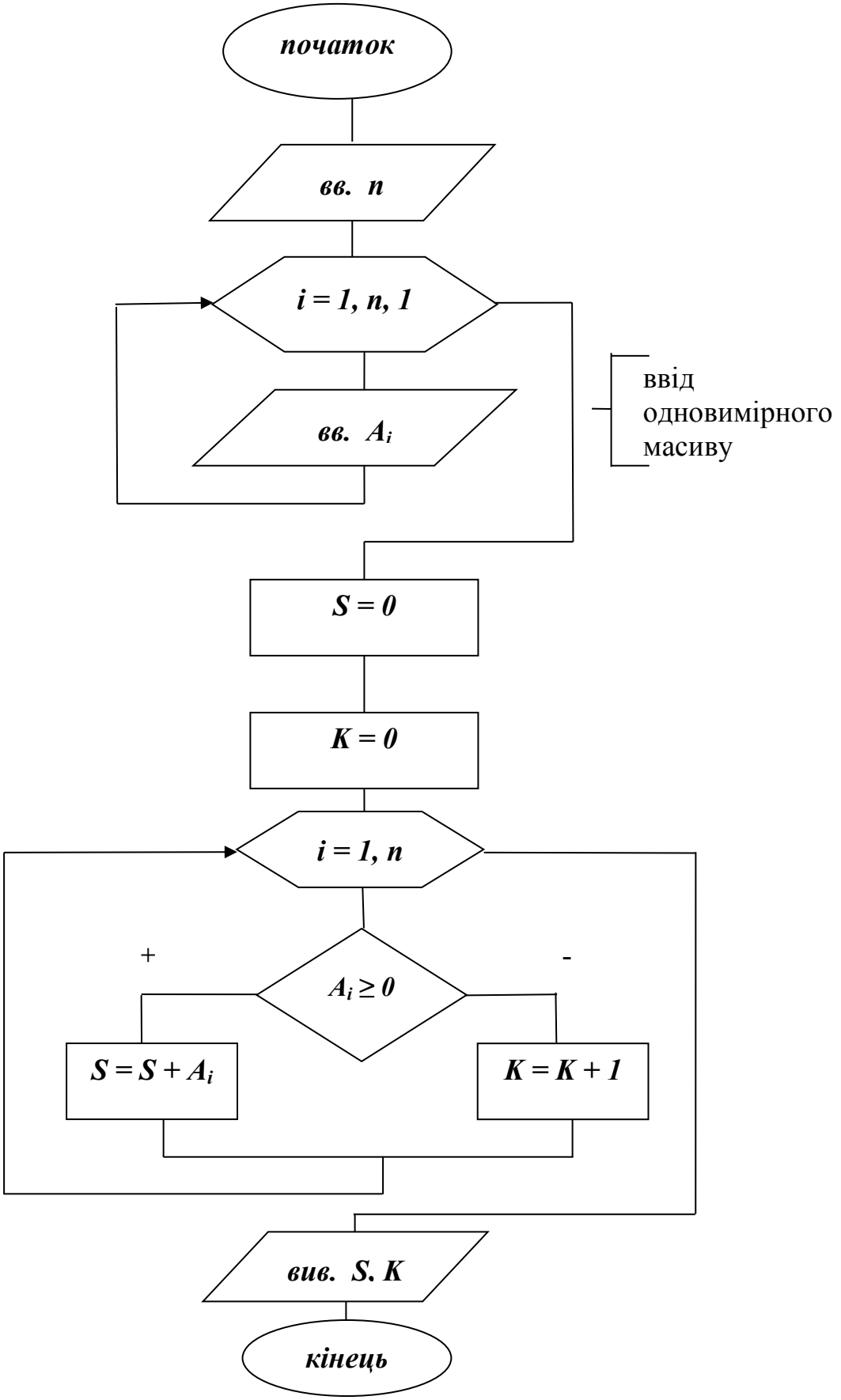
1. Перетин діапазонів розгалуження має утворювати порожню множину.
2. Об'єднання діапазонів розгалуження має співпадати з областю допустимих значень параметрів розгалуження.

$$A = \begin{cases} f_1(x, y) & , x < y \\ f_2(x, y) & , x \geq y \end{cases}$$



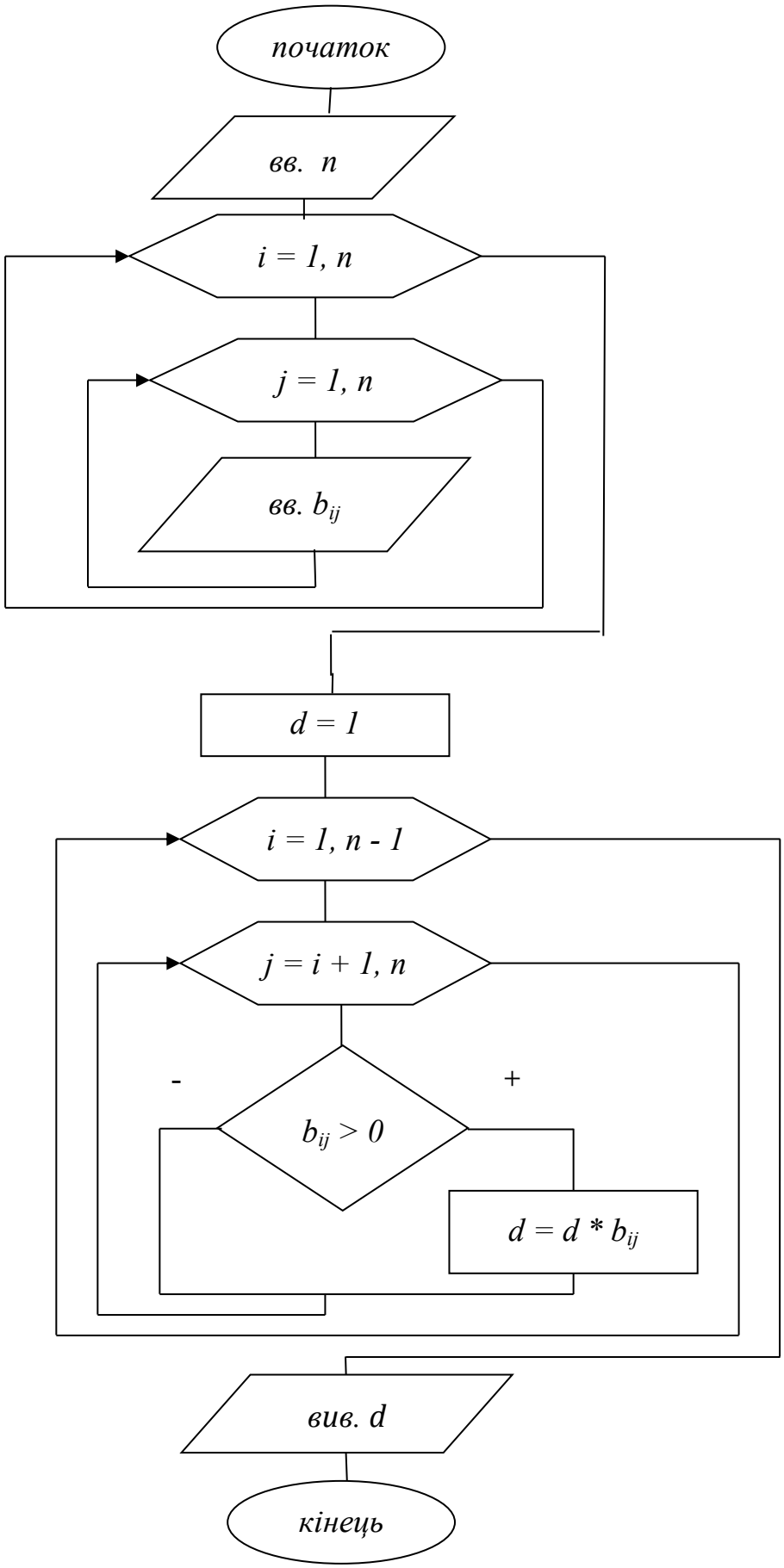
Одновимірний масив.

Для заданого масиву **A** підрахувати суму додатних та кількість від'ємних елементів.



Обробка двохвимірного масиву.

Знайти добуток додатних елементів що розміщені над головною діагоналлю квадратної матриці.



Завдання 1.

Скласти блок-схему алгоритму розгалуженого обчислювального процесу

№ п/п	Завдання
1	$A=\begin{cases} e^{xy}+D\ln x, & якщо\ \ x<y \\ \sqrt[3]{x+C}+\ln y, & якщо\ x\ge y \end{cases}$
2	$A=\begin{cases} \lg(Cx+y), & якщо\ \ x\le y \\ \sqrt[3]{x}+De^{y}, & якщо\ \ x>y \end{cases}$
3	$A=\begin{cases} tgC+xe^{y}, & якщо\ \ x>y \\ \lg y+Dx, & якщо\ \ x\le y \end{cases}$
4	$A=\begin{cases} \lvert x+y\rvert+Ce^{x}, & якщо\ x\ge y \\ \ln(xy)+D, & якщо\ x<y \end{cases}$
5	$A=\begin{cases} C\sqrt[3]{x+y}, & якщо\ x<y \\ De^{xy}, & якщо\ x\ge y \end{cases}$
6	$A=\begin{cases} \ln Cx+y^{2}, & якщо\ x<y \\ \sqrt[3]{Dy}+e^{x}, & якщо\ x\ge y \end{cases}$
7	$A=\begin{cases} \ln\sqrt{x}+Cy, & якщо\ x\le y \\ tgxy+D\sqrt[3]{x}, & якщо\ x>y \end{cases}$
8	$A=\begin{cases} \cos 2x+3Cx^{2}, & якщо\ x>y \\ Dxy+\lg\lvert x+y\rvert, & якщо\ x\le y \end{cases}$
9	$A=\begin{cases} Ce^{xy}+\sqrt[3]{x}, & якщо\ x\ge y \\ \ln Dx, & якщо\ x<y \end{cases}$
10	$A=\begin{cases} C\lvert xy\rvert+\ln x, & якщо\ x>y \\ De^{\sqrt{x}}+\sqrt[3]{y}, & якщо\ x\le y \end{cases}$
11	$A=\begin{cases} e^{x}+C\ln(x^{2}), & якщо\ x>y \\ \sqrt[3]{xy}+Dy, & якщо\ x\le y \end{cases}$
12	$A=\begin{cases} C\ln xy, & якщо\ x<y \\ \sqrt[3]{(x+y)^{C}}, & якщо\ x\ge y \end{cases}$
13	$A=\begin{cases} D\ln x^{2}, & якщо\ x<y \\ C\sin y+e^{x}, & якщо\ x\ge y \end{cases}$
14	$A=\begin{cases} Ce^{xy}+\lg x, & якщо\ x<y \\ tg\,xy+Cy^{2}, & якщо\ x\ge y \end{cases}$

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

15	$A=\begin{cases}\sqrt{xy}+D\ln y, & якщо\ x>y\\ Cx^2+Dy, & якщо\ x\le y\end{cases}$
16	$A=\begin{cases}C^{xy}+\ln x, & якщо\ x<y\\ \cos x+\ln y, & якщо\ x\ge y\end{cases}$
17	$A=\begin{cases}xy+tgx, & якщо\ x\ge y\\ \sqrt[3]{x}+Dy^2, & якщо\ x<y\end{cases}$
18	$A=\begin{cases}\ln(xy)+yx, & якщо\ x\le y\\ \lvert x-y\rvert-Ce^y, & якщо\ x>y\end{cases}$
19	$A=\begin{cases}e^y+\sqrt{Cx+y}, & якщо\ x>y\\ \sqrt[3]{x}+Dy, & якщо\ x\le y\end{cases}$
20	$A=\begin{cases}Dtgx+xy, & якщо\ x\le y\\ e^x+\ln(xy), & якщо\ x>y\end{cases}$
21	$A=\begin{cases}Ce^{\sqrt{x}}+Dy, & якщо\ x\ge y\\ C\lvert xy\rvert+y, & якщо\ x<y\end{cases}$
22	$A=\begin{cases}x+\sqrt[3]{y}, & якщо\ x<y\\ \cos Cy-Dx, & якщо\ x\ge y\end{cases}$
23	$A=\begin{cases}xy-Cy\sqrt{y}, & якщо\ x\le y\\ e^x y^2+D, & якщо\ x>y\end{cases}$
24	$A=\begin{cases}Dx+Cy^2, & якщо\ x<y\\ \sqrt[3]{Dx+C}+\ln y, & якщо\ x\ge y\end{cases}$
25	$A=\begin{cases}Cxy-Dx\sqrt{y}, & якщо\ x\ge y\\ \sqrt[3]{xy}+Dy, & якщо\ x<y\end{cases}$
26	$A=\begin{cases}e^{xy}+Dx, & якщо\ x<y\\ Cx+\ln y, & якщо\ x\ge y\end{cases}$
27	$A=\begin{cases}e^y+\sqrt[3]{x}, & якщо\ x\le y\\ tgD+\ln y, & якщо\ x>y\end{cases}$
28	$A=\begin{cases}C\lvert x-y\rvert+\ln x, & якщо\ x\le y\\ e^{Cx}-\sqrt[3]{y}, & якщо\ x>y\end{cases}$
29	$A=\begin{cases}e^{Dx}+\ln y, & якщо\ x<y\\ \sqrt[3]{Cxy}+D\ln y, & якщо\ x\ge y\end{cases}$

30	$A=\begin{cases} D\ln x - e^{y}, & якщо\ x < y \\ \sqrt[3]{x+Cy} + \ln y, & якщо\ x \geq y \end{cases}$

Завдання 2.

Скласти блок-схему алгоритму обробки одновимірного масиву:

1. Дано масив А(15). Знайти добуток від`ємних елементів масиву.
2. Дано масив А(15). Знайти кількість елементів, більших заданого числа В.
3. Дано масив А(15). Знайти кількість елементів, менших заданого числа В.
4. Дано масив А(15). Знайти кількість елементів, рівних заданому числу В.
5. Дано масив А(15). Знайти максимальний елемент масиву.
6. Дано масив А(15). Знайти мінімальний елемент масиву.
7. Дано масив А(15). Знайти номери всіх додатних елементів.
8. Дано масив А(15). Знайти номери всіх від`ємних елементів.
9. Дано масив А(15). Знайти добуток елементів з парними номерами.
10. Дано масив А(15). Розділити всі елементи на найменший елемент масиву (відмінний від нуля).
11. Дано масив А(15). Розділити всі елементи на найбільший елемент масиву (відмінний від нуля).
12. Дано масив А(15). Знайти суму елементів з непарними номерами.
13. Дано масив А(15). Знайти суму елементів з парними номерами.
14. Дано масив А(15). Знайти добуток елементів з непарними номерами.
15. Дано масив А(15). Знайти кількість додатних елементів масиву.
16. Дано масив А(15). Знайти кількість від`ємних елементів масиву.
17. Дано масив А(15). Знайти суму додатних елементів масиву.
18. Дано масив А(15). Знайти суму від`ємних елементів масиву.
19. Дано масив А(15). Знайти середнє арифметичне додатних елементів масиву.
20. Дано масив А(15). Знайти середнє арифметичне від`ємних елементів масиву.
21. Дано масив А(15). Знайти мінімальний додатній елемент масиву.
22. Дано масив А(15). Знайти максимальний від`ємний елемент масиву.
23. Дано масив А(15). Знайти кількість елементів, рівних нулю.
24. Дано масив А(15). Знайти добуток додатних елементів масиву.
25. Дано масив А(15). Поміняти місцями найбільший елемент з останнім.
26. Дано масив А(15). Поміняти місцями найменший елемент з останнім.
27. Дано масив А(15). Замінити від`ємні елементи масиву найбільшим елементом.
28. Дано масив А(15). Замінити додатні елементи масиву найбільшим елементом.

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

29. Дано масив A(15). Замінити від`ємні елементи масиву найменшим елементом.
30. Дано масив A(15). Замінити додатні елементи масиву найменшим елементом.

Завдання 3.

Скласти блок-схему алгоритму обробки двохвимірного масиву:

1. Дано масив C(5,4). Знайти середнє арифметичне елементів кожної стрічки.
2. Дано масив H(3,5). Знайти суму від'ємних елементів кожної стрічки.
3. Дано масив T(6,4). Знайти максимальний елемент кожної стрічки.
4. Дано масив M(2,6). Знайти мінімальний елемент кожного стовпця.
5. Дано масив E(4,5). Для кожної стрічки знайти суму елементів більших ***10***.
6. Дано масив AD(3,5). Знайти добуток всіх елементів з непарних стовпців.
7. Дано масив KD(4,3). Знайти суму максимальних елементів всіх стрічок.
8. Дано масив MD(4,6). Знайти кількість від'ємних елементів кожної стрічки.
9. Дано масив OD(5,3). Знайти кількість додатних елементів кожного стовпця.
10. Дано масив HD(4,4). Для кожного стовпця знайти кількість елементів більших по модулю числа **π.**
11. Дано масив D(4,4). Знайти добуток елементів діагоналей матриці.
12. Дано масив KD(4,4). Знайти суму елементів що лежать над головною діагоналлю.
13. Дано масив OD(5,3). Всі додатні елементи замінити їхнім логарифмом.
14. Дано масив E(4,5). Всі від'ємні елементи замінити їхнім модулем.
15. Дано масив KD(4,3). Всі елементи кожної стрічки розділити на перший ненульовий елемент стрічки.
16. Дано масив A(5,5). Знайти суму елементів масиву.
17. Дано масив B(5,6).Знайти добуток елементів масиву.
18. Дано масив X(6,8). Знайти значення найбільшого елементу масиву.
19. Дано масив Y(4,5). Знайти значення найменшого елементу масиву.
20. Дано масив Z(6,3). Знайти кількість додатних елементів масиву.
21. Дано масив B(6,6). Знайти добуток відмінних від нуля елементів.
22. Дано масив C(3,5). Всі елементи розділити на найбільший, відмінний від нуля, елемент.
23. Дано масив D(3,4) Всі елементи помножити на найменший, відмінний від нуля, елемент.
24. Дано масив KD(4,4). Знайти суму елементів кожної з діагоналей.
25. Дано масив KD(4,4). Знайти добуток елементів що лежать нижче головної діагоналі.

26. Дано масив D(4,4). Знайти суму елементів кожної стрічки.
27. Дано масив A(3,5). Знайти суму додатних елементів кожної стрічки.
28. Дано масив P(4,3). Знайти добуток елементів кожної стрічки.
29. Дано масив K(4,3). Знайти добуток додатних елементів кожного стовпця.
30. Дано масив D(3,7). Для кожної стрічки знайти суму елементів непарних стовпців.

Зауваження:

В процесі виконання роботи студент веде записи в робочому зошиті. По завершенню оформляє звіт, в якому також дає відповіді на контрольні питання за вибором викладача.

Контрольні питання:

1. Поняття алгоритму. Основні типи алгоритмів.
2. Лінійний обчислювальний процес. Приклади.
3. Розгалужені обчислювальні процеси. Приклади.
4. Вимоги до розгалужених обчислювальних процесів.
5. Циклічні обчислювальні процеси. Приклади.
6. Вимоги до циклічного обчислювального процесу.
7. Опис алгоритму обчислення значень функції на заданому проміжку.
8. Робота блоку модифікації.
9. Обчислення скінчених сум.
10. Обчислення скінчених добутків.
11. Організація вводу елементів одновимірного масиву.
12. Організація вводу елементів двовимірного масиву.

Лабораторна робота №2
(2 год.)

Тема: Порядок реалізації Сі-програм в діалоговому режимі на персональних ЕОМ. Розв'язування задач з лінійним обчислювальним процесом.

Мета роботи: Освоїти роботу в інтегрованому середовищі С++. Оволодіти практичними навичками розробки та програмування обчислювального процесу лінійної структури. Засвоїти запис арифметичних виразів, команд присвоєння та вводу/виводу. Навчитись вводити, записувати і редагувати текст програми; здійснювати процес компіляції, виправлення можливих помилок та виконання програми.

За час виконання лабораторної роботи студент повинен освоїти:

- Використання в програмах констант, змінних, стандартних функцій.
- Правила запису арифметичних виразів.
- Застосування арифметичної команди присвоєння.

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

- Організацію вводу вхідних даних з використанням комбінації команд виводу і вводу або через опис відповідних констант.
- Організацію виводу результатів.
- Послідовність дій при виклику інтегрованого середовища С++.
- Основні прийоми роботи по вводу та редагуванню тексту програми.
- Порядок компіляції, виправлення можливих помилок та виконання програми**.**
- Запис тексту програми на магнітний диск та виклик його в вікно редактора.

Завдання на лабораторну роботу: Скласти схему алгоритму та програму на С++ обчислення значень виразів для заданих значень вхідних величин.

Порядок виконання лабораторної роботи:

- Завантажити інтегроване середовище С++ (***Студенти можуть використовувати довільні середовища програмування С++ та працювати на власних комп'ютерах);***
- Описати загальний вигляд головного меню та послідовність переходу в вікно редактора;
- Ввести текст програми (в першій стрічці обов'язково має бути коментар, в якому вказати номер роботи, групу та прізвище виконавця). Ввід вхідних даних має бути організований з використанням комбінації операторів виводу і вводу.
- Текст програми записати у власну папку у виді: (**Prizvyshche_L2v30**).
- Відкомпілювати програму, виправляючи при цьому можливі помилки.
- Відлагоджену програму виконати, записуючи отримані результати.
- Оформити звіт про виконану роботу.

Зауваження: Імена створюваних програмних файлів повинні обов'язково містити прізвище, абревіатуру лабораторної роботи і порядковий номер варіанту.

Теоретичні відомості

Програми складаються із синтаксичних конструкцій, які називаються командами (інші назви – оператори, вказівки, речення). Команди будуються з лексем – неподільних елементів мови: слів, чисел, символів операцій.

Загальна структура програми:

```
//коментарі
#include <назва бібліотечного файлу 1>
#include <назва бібліотечного файлу N>
<інші директиви препроцесора>
using  namespace std;
<оголошення глобальних змінних>;
<оголошення глобальних сталих>;
```

```
<оголошення та створення функцій користувача>
…
<тип результату функції> main(опис формальних параметрів)
{
<оголошення локальних змінних>;
<оголошення локальних сталих>;
<команди>;
}
```

Директиви препроцесора. Препроцесор - це програма, яка опрацьовує директиви. Директиви препроцесора – це команди компілятора, які виконуються до початку компіляції програми. Директиви мови С++ починаються із символу #. Розглянемо декілька типів директив.

Директива ***include*** означає, що до програми необхідно приєднати програмний код із зазначеного після неї файлу. Наприклад

```
#include <cmath>
```

Директива #define має подвійне значення. По-перше, вона може задати стале значення (оголошує сталу). Наприклад, якщо у програмі задано

```
#define N 25
```

то N під час виконання програми матиме значення 25. По-друге, вона дає змогу описати ***макроси*** - короткі команди (переозначити команди) чи записати функції, наприклад, так:

```
#define D(a, b, c) (b * b - 4 * a * c)
```

Тепер скрізь для обчислення дискримінанта замість команди

```
d=b*b-4*a*c
```

можна записувати

```
d = D(a, b , c)
```

Розглянемо програму, у результаті виконання якої на екран буде виведено повідомлення***: Привіт, студенте! Я С++!***

```
// Моя перша програма мовою С++
#include <iostream>
  using  namespace std;
  int main()
      {
      cout << " Привіт, студенте! Я С++!";
      return 0;
      }
```

(тут і надалі зразки та фрагменти програм написані на С++ та реалізовані в середовищі Dev-C++)

Розглянемо елементи програми. У першому рядку є коментар. ***Коментар*** - це фрагмент тексту програми, який слугує для пояснення призначення програми чи окремих команд і не впливає на виконання команд. Його записують так: ***//текст коментаря*** або так: ***/* текст коментаря */.***

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

У першому випадку коментар має бути або у кінці рядка, або єдиним у рядку. Другий спосіб більш універсальний: коментар можна записувати будь-де, не розриваючи лексем.

Директива **#include** <***iostream***> під'єднує бібліотечний файл ***iostream***. Саме у цьому файлі описані функції, які дають змогу виконувати операції введення-виведення даних.

Далі у програмі записана обов'язкова функція ***main().*** Ключове слово ***int*** означає, що функція ***main()*** повертатиме у точку виклику результат цілого типу.

Конструкція ***cout*** << забезпечує виведення на екран монітора повідомлення "***Привіт, студенте! Я С++!".***

Команда ***return*** слугує для виходу з функції ***main().*** Числовий параметр після ***return*** є результатом (значенням) функції (у цій програмі - **0**).

Цілі типи. Цілі типи описані в табл. 1.

Таблиця 1. Дані цілочисельних типів

Назва типу	Обсяг, байтів	Діапазон допустимих значень
int	2 або 4	-32768 ... 32767 або -2147483648 ... 2147483647
short int	2	-32768 ... 32767
unsigned short int	2	0 ... 65535
long int	4	-2147483648 ... 2147483647
unsigned long int	4	0 ... 4294967295

Оголосимо три змінні цілого типу:

int x, y; short int z;

Дійсні типи. Дійсні типи описані в табл. 2.

Таблиця 2. Дійсні типи

Назва типу	Обсяг, байтів	Діапазон значень
float	4	$\pm 3.4*10^{-38}$ ...$\pm 3.,4*10^{38}$
double	8	$\pm 1.7*10^{-308}$ ...$\pm 1.7*10^{308}$
long double	10	$\pm 1.18*10^{-4932}$ ...$\pm 1.8*10^{4932}$

В описах змінних та констант можна задавати їх значення:

float h, pi = 3.1415926;
double v = 365.976;
const float w = -12, h = 23.4;

Команда присвоєння має такий загальний вигляд:

<назва змінної> = <вираз>

або

<назва змінної1> = <назва змінної2> = ... = <назва змінноїN> = <вираз>

Дія команди. Обчислюється вираз і його значення надається змінній або декільком змінним. ***Вираз*** призначений для описування формул, за якими

виконуватимуться обчислення. Вираз може містити числа, сталі, змінні, назви функцій, з'єднані символами операцій.

Приклади:

a = 8-2;
c = d = a + 4;
e = d / 5 + c;

Основні арифметичні операції мови С++ наведені в табл. 3.

Таблиця 3. Арифметичні операції

Пріоритет	**Операції**	**Зміст операції**
1	+, -	Присвоєння знака
2	*,/,%	Множення, ділення, остача
3	+, -	Додавання, віднімання
4	==, !=, <, <=, >, >=	Порівняння (відношення)

Операції інкременту (збільшення)(++) та декременту (зменшення) (--).

Операції інкременту і декременту існують у двох формах – *префіксній* та *постфіксній.* Якщо символи ++ (--) записані перед змінною – то це інкремент (декремент) у префіксній формі, а якщо після змінної – у постфіксній. Операція інкременту має такий вигляд:

++<змінна> ***або*** ***<змінна>++***

Дія операції. Значення змінної збільшується на одиницю. Команди **++a, a++** рівносильні команді ***a = a + 1.*** Форма інкременту (декременту) впливає на порядок виконання операцій у виразах. Розглянемо це на прикладах.

a = 2;
b = 3 * ++a;

Результати будуть такими: ***a = 3, b = 3 * 3 = 9***. Тут використано операцію інкременту у префіксній формі: спочатку збільшується значення змінної *a* на одиницю, а пізніше обчислюється вираз.

c = 5;
d = (c++) + 4;

Тут спочатку обчислюється вираз (для ***d***) з *c* = 5, а потім збільшується значення змінної с на одиницю. Тобто ***d = c + 4 = 5 + 4 = 9, c = c + 1 = 5 + 1 = 6*** (це операція інкременту у постфіксній формі). Оскільки у виразі записано три знаки плюс "+" підряд, то для однозначного визначення порядку операцій використано круглі дужки.

Аналогічно операція декременту має такий вид:

--<змінна> або ***<змінна>--***

Значення змінної зменшується на одиницю. Команди --a та a-- діють як і команда ***a = a - 1.***

Приклад

x = 4;
y=15/--x;

Результати будуть такими: **x = 3, y = 15 / 3 = 5.** Тут спочатку значення змінної *x* зменшується на одиницю, а пізніше обчислюється вираз для *y.*

Приклад

f = 20;

g = (f--)-10;

змінним **f, g** будуть надані значення 19 і 10 відповідно.

Команда присвоєння, суміщена з арифметичною операцією. Щоб надати значення змінній, можна скористатися командою присвоєння, суміщеною з деякою арифметичною операцією, а саме: +=, -=, *=, /=, %=. Загальний вигляд команди присвоєння, суміщеної з арифметичною операцією, такий:

<змінна> <символ арифметичної операції>= <вираз>

Зауважимо, що між символом арифметичної операції та символом "=" пропуск не допускається (***це лексема***). Наприклад, команди

a = a+10 ***та*** ***a += 10***

рівносильні. Виконавши їх одержимо однаковий результат, хоча оптимальніше використовувати а ***+= 10***. А замість команди ***b = b * 4*** можна писати ***b *= 4.*** Аналогічно, замість ***d = d / 3*** запишемо ***d /= 3.***

Математичні функції.

Усі стандартні математичні функції у С++ описані у бібліотеці ***math.h.*** Тому, якщо вони використовуються, на початку програми необхідно записати рядок під'єднання потрібного файлу заголовку

#include <cmath>;.

Основні математичні функції бібліотеки ***math*** наведені у табл. 4. Аргументи функцій записують у круглих дужках.

Таблиця 4. Математичні функції

Транскрипція запису С++	**Позначення**	**Назва**
abs(x)	$\lvert x \rvert$	абсолютне значення цілого числа (модуль)
fabs(x)	$\lvert x \rvert$	абсолютне значення дійсного (float) числа (модуль)
labs(x)	$\lvert x \rvert$	модуль довгого цілого
acos(x)	$\arccos(x)$	арккосинус
asin(x)	$\arcsin(x)$	арксинус
atan(x)	$arctg(x)$	арктангенс
cos(x)	$\cos(x)$	косинус
sin(x)	$\sin(x)$	синус
tan(x)	$tg(x)$	тангенс
exp(x)	e^x	експонента
log(x)	$\ln(x)$	логарифм натуральний
log10(x)	$\lg(x)$	логарифм десятковий
pow(a,b)	a^b	степінь
pow10(b)	10^b	степінь десяти
sqrt(x)	$\sqrt{x}$	квадратний корінь
cbrt(x)	$\sqrt[3]{x}$	кубічний корінь
pow(x,1.0/n)	$\sqrt[n]{x}$	корінь n-го степеня

floor(x)		відкидає дробову частину числа *x*
fmod(x, y)		обчислює остачу від ділення числа *x* на число *y*

Усі наведені функції, крім ***abs(x)***, мають тип аргументу і результату ***double.*** Для функції ***abs(x)*** типом аргументу і результату є ***int.***

Зауваження. Замість оголошення сталої ***const float pi = 3.1415926*** можна використовувати стандартну сталу ***M_PI***, яка описана у модулі ***cmath.***

Введення-виведення даних

У С++ немає вбудованих команд уведення-виведення даних. Для організації введення-виведення тут реалізована концепція ***потоків***, яка визначена у спеціальних модулях. У модулі ***istream*** описані команди введення, у модулі ***ostream*** – команди виведення, а у модулі ***iostream*** – команди виведення і введення

Під потоком розуміють процес уведення-виведення інформації у файл. Периферійні пристрої введення-виведення, такі як клавіатура, монітор, принтер, розглядаються як текстові файли. Під час виконання будь-якої програми автоматично підключаються стандартні потоки для введення даних з клавіатури (***cin***), виведення на екран (***cout***).

Стандартні потоки використовують команди введення (>>) та виведення (<<) даних. За замовчуванням стандартним пристроєм для потоків виведення даних і повідомлень про помилки є монітор користувача, а для потоку введення даних – клавіатура. Однак потоки можна перенаправляти, наприклад, можна зчитувати вхідну інформацію для програми не з клавіатури, а з деякого текстового файлу на диску.

Команда введення даних. Надавати значення змінним можна двома способами: за допомогою команди присвоєння, наприклад **x = 3.1,** або команди ***уведення даних із клавіатури***. Команда уведення даних із клавіатури дає змогу виконувати програму для різних вхідних даних, що робить її більш універсальною (масовою). Команда введення >> описана у бібліотеці iostream (istream) і має такий загальний вигляд

cin >> <змінна 1> >> <змінна 2> >> ... >> <змінна N>;

Дія команди. Виконання програми зупиняється. Система переходить у режим очікування введення даних. Користувач набирає на клавіатурі значення змінних через пропуск і натискає на клавішу вводу. У результаті виконання цієї команди змінним буде присвоєне конкретне значення (те, яке користувач уведе з клавіатури).

Приклад

```
int a, b, c;
cin >> a >> b >> c;
```

Під час виконання програми на клавіатурі набираємо

3 4 5 натискаємо на клавішу ***Enter***

У результаті виконання команд змінні набувають таких значень:

a = 3, b = 4, c = 5.

Команда виведення даних. Команда виведення << описана у бібліотеці ***iostream (ostream)*** і має такий загальний вигляд

cout << <вираз1> << <вираз2> <<…<< <виразN>;

У списку можуть бути константи, змінні чи вирази. Текст у потоці виводу записують у лапках, які на пристрій виводу не виводяться

Приклад

cout << "площа = "<<s;

Для того, щоб вивід здійснювався у потрібному вигляді, використовують керуючі послідовності.

Послідовності символів, які починаються з символу \ (обернений слеш) називаються керуючими або ***escape***-послідовностями (таблиця 5).

Таблиця 5 – ***Escape-послідовності***

Спеціальний символ	Значення
\a	звуковий сигнал
\b	повернення на 1 символ
\f	переведення сторінки
\n	перехід на наступний рядок
\r	повернення каретки
\t	горизонтальна табуляція
\v	вертикальна табуляція
\\	символ \
\'	символ '
\"	символ "
\?	символ ?
\0	нульовий символ

Керуючі послідовності разом з пояснюючим текстом записують в лапках.

Приклад

```
cout << "Увага! \n";
cout << "Введіть вхідні дані \n";
```

Замість **"\n"** можна використовувати ***endl*** – кінець рядка.

Як правило, при введені вхідних даних перед командами вводу записують команди виводу певної підказки. Наприклад:

```
cout << "a= "; cin >> a;
cout << "b= "; cin >> b;
cout << "c= "; cin >> c;
```

Як приклад до даної лабораторної роботи запишемо програму лінійного обчислювального процесу обчислення значень двох виразів для заданих значень змінних

$q=\frac{\lvert d+f\rvert+tg(f^2+h^2)}{k+4\ln f}$ $\quad u=\frac{h+\sin^2 k+\cos h}{e^f+d\ln(h+k)}$	h=0.1 k=8.3 d=4.9 f=0.8

```
// Лабораторна робота №2 варіант 31 група СІ-11 Коломієць М.В.
#include<iostream>
#include<cmath>
using namespace std;
float k,f,h,d,q,u;
int main()
{
system("color F0"); // Встановити білий фон і чорний текст
cout<<"h=";cin>>h;
cout<<"k=";cin>>k;
cout<<"d=";cin>>d;
cout<<"f=";cin>>f;
q = (fabs(d + f) + tan(f * f + h * h)) / (k + 4 * log(f));
u = (h + pow(sin(k),2) + cos(h)) / (exp(f) + d * log(h + k));
cout<<"q= "<<q<<endl<<"u= "<<u<<endl;
return 0;
}
```

Варіанти завдань лабораторної роботи

Варіант завдання	Розрахункові вирази	**Вхідні дані**
1	$d=1+\frac{z^2}{3+z^2/5}$ $\quad c=\frac{2\cos(s-\pi/6)}{1/2+\sin^2 z}$	s=1.426 z=-1.22
2	$g=\left\lvert c^{d/s}-\sqrt{d/s}\right\rvert$ $\quad f=(c-d)\frac{c+s/(d-c)}{1+(d-s)^2}$	c=1.825 d=18.2 s=3.2
3	$y=\frac{e^{4.7\sqrt[3]{ax+b^2}}-\lg(bx)}{tg(bx)+\cos(b)}$ $\quad z=bx^{7.1}\lg(ax)+\frac{e^{ab}tg^2(ax)}{\cos(bx)}$	a=6.3 b=2.1 x=0.5
4	$d=e^{-kc}\sin(t+c)-\sqrt{\lvert kt+c\rvert}$ $\quad s=c\sin(kt^2\cos 2t)-1$	k=-0.5 c=1.7 t=0.44
5	$w=\sqrt{n^2+m}-n^2\sin^2(n+k)/n$ $\quad h=\cos^2 n^2-n/\sqrt{k^2+m^2}$	k=1.5 m=15.5 n=2.9
6	$s=f^2tg^2(f+k)^2+t/\sqrt{f+k}$ $\quad Q=\frac{tf^2-k}{e^{ft}-1}$	t=16.5 k=3.4 f=0.61

7	$R = f^2(f+1)/k - \sin^2(f+h)$ $s = \sqrt{fk/h} + \cos^2(f+k)^2$	h=0.7 k=0.05 f=0.5
8	$w = \sin^2(f^2+h)^2 - \sqrt{f/k}$ $z = \frac{f^2}{h} + \cos(f+k)^2$	h=1.1 k=0.04 f=0.2
9	$f = \sqrt{m\,tg\ t + \lvert c\sin t\rvert}$ $z = m\cos(kt\sin t) + c$	m=2 c=-1 t=1.2 k=0.7
10	$w = k\,tg^2 f - \frac{h}{\sin^2(f/h)}$ $d = he^{-f}\cos(kf/h)$	h=3.2 k=17.5 f=-4.8
11	$w = \ln(h+f^2) + \sin^2(f/k)$ $z = e^{-cf}\,\frac{f+\sqrt{f+h}}{f-\sqrt{\lvert f-c\rvert}}$	h=10.2 k=9.2 f=2.2 c=0.5
12	$w = \frac{h^{kf} + k^{-f}\cos(h+k)f}{f-1}$ $R = \sqrt{f^2+k} - k^2\sin^2(f+h)/f$	h=0.3 k=0.9 f=0.61
13	$z = \sqrt{hf\sin(2f) + e^{-2f}(f+k)}$ $w = \cos^2 f^2 - f/\sqrt{h^2+k^2}$	h=0.5 k=3.1 f=1.4
14	$U = \frac{h^2 f + e^{-f}\cos kf}{fk - e^{-f}\sin fk + 1}$ $g = e^{hk}\ln(h+f) - e^f\ln(k-f)$	h=0.5 k=2.9 f=0.3
15	$z = \frac{\sin f}{\sqrt{1+m^2\sin^2 f}} - cm\ln(mf)$ $s = e^{-mf}\sqrt{f+1} + e^{-kh}\sqrt{h+1.5}$	m=0.7 c=2.1 f=1.7 h=0.5 k=1.08
16	$w = \frac{fk+\sqrt{c}}{\sin f + \ln k}$ $d = e^k + hf^2 + tgk$	k=1.2 c=2.4 f=0.2 h=1.2
17	$w = \sqrt{r^2 + k\ln c} + tg^2 h$ $z = r^2 + \cos^2 h + e^k$	h=0.2 k=5.3 c=7.4 r=2.8
18	$w = \ln c + f^2 d - ktgf$ $l = \frac{e^2 + d\sin f}{\cos k + c^x}$	k=6.4 c=5.6 d=3.9 f=6.9
19	$w = \frac{d^2 f - \sin(hf) + tgh}{f + h\sqrt{d}}$ $c = \ln h + d^f - h$	h=4.2 d=2.8 f=2.7
20	$t = h(\sin^2 h + c\cos^2 k)$ $z = h\ln(h^k + c) + 3$	h=7.3 k=3.5 c=2.5

21	$w=(h+f)\dfrac{\sin(h+fk)}{2+f^2}$ $z=(k^2+h^2f)-3\ln(hk)$	h=4.1 k=7.2 f=5.2
22	$k=\ln(df+1)+\sin^2(cf)$ $l=tg(d+cf^2)-2e^d$	d=4.8 c=6.5 f=4.1
23	$s=\sqrt{kw+d^2}+\cos\dfrac{k}{d}$ $r=\dfrac{(b+w)^2}{d\ln k}$	k=9.7 d=2.9 w=3.0
24	$p=\dfrac{\sin^2(h+kf)}{k^2+hf+\ln f}$ $t=\sqrt{h^2+e^k}+tg^2 f$	h=5.8 k=3.9 f=8.1
25	$w=\dfrac{d^2+ftg(c+1)}{\ln(d+f)+c^2}$ $z=\dfrac{e^d+\ln c}{f+\sqrt{cd+f}}$	c=1.5 d=3.9 f=0.3
26	$m=hf^2+\dfrac{tg(k+f)}{e^k+3}$ $n=\sqrt{h^2+f}-\lvert\sin k\rvert+\ln(h+k)$	h=2.9 k=5.3 f=8.1
27	$p=e^{-d}\sqrt{f^2+\ln\dfrac{d}{f}}+k$ $r=\dfrac{\cos^2(k+df)+3}{f^2+k\ln(f+d)}$	k=9.1 d=7.7 f=8.9
28	$w=tg^2(h+f)+\ln k+e^k$ $z=\dfrac{\sqrt{h+(cf+k)^2}}{\sin^2(k+c)+2}$	h=3.1 k=0.4 c=3.2 f=4.2
29	$j=\dfrac{\sqrt{m+kf}+lf+tgm}{k^2+\sin m}+\ln f$ $w=e^k+k(m^2+lf)$	k=4.8 l=2.3 m=4.1 f=1.4
30	$q=\dfrac{\lvert d+f\rvert+tg(f^2+h^2)}{k+4\ln f}$ $u=\dfrac{h+\sin^2 k+\cos h}{e^f+d\ln(h+k)}$	h=0.1 k=8.3 d=4.9 f=0.8

Контрольні питання:

1. Головне меню інтегрованого середовища Dev-C++.
2. Використання функціональних клавіш в інтегрованому середовищі Dev-C++.
3. Послідовність вводу тесту програми в Dev-C++.
4. Призначення основних пунктів меню File інтегрованого середовища Dev-C++.
5. Послідовність першого запису тексту програми на диск в Dev-C++.
6. Основні стандартні математичні функції в C++.
7. Арифметичні вирази. Правила їх запису в C++.
8. Призначення та структура опису міток в C++.
9. Призначення та структура опису констант в C++.
10. Призначення та структура опису змінних в C++.
11. Команда присвоєння в C++. Синтаксис та порядок роботи.
12. Команда вводу в C++. Синтаксис та порядок роботи.
13. Команда виводу в C++. Синтаксис та порядок роботи.

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

14. Організація діалогу при вводі значень змінних в С++.
15. Виклик тексту програми з диску в Dev-C++.
16. Виправлення можливих помилок в процесі компіляції програми в Dev-C++.
17. Основні прийоми редагування тексту програми в Dev-C++.
18. Послідовність компіляції програми в в інтегрованому середовищі Dev-C++.
19. Встановлення шрифту кирилиці в редакторі інтегрованого середовища Dev-C++.

Лабораторна робота №3
(2 год.)

Тема: Реалізація на ЕОМ програм з розгалуженим обчислювальним процесом.

Мета роботи: Оволодіти практичними навиками розробки та програмування обчислювального процесу розгалуженої структури. Засвоїти запис логічних виразів, умовних та складених команд, команди **?**.

За час виконання лабораторної роботи студент повинен освоїти:

- Використання в програмах констант, змінних, стандартних функцій.
- Правила запису логічних виразів.
- Застосування умовної команди.
- Застосування складеної команди.
- Застосування команди **?.**
- Організацію вводу вхідних даних з використанням комбінації команд виводу і вводу або через опис відповідних констант.

Завдання на лабораторну роботу: Скласти схему алгоритму та три програми на С++ обчислення значень виразів розгалуженої структури (із використанням умовної команди, складеної команди або команди ***?***) для заданих значень вхідних величин.

Константи *C=2.1, D=0.56*.

Порядок виконання лабораторної роботи:

- Завантажити інтегроване середовище С++;
- Ввести текст програми (в першій стрічці обов'язково має бути коментар, в якому вказати номер роботи, групу та прізвище виконавця). Ввід вхідних даних має бути організований з використанням комбінації команд виводу і вводу.
- Текст програми записати у власну папку у виді: (***Prizvyshche_lr3v31_1***).
- Відкомпілювати програму, виправляючи при цьому можливі помилки.
- Відлагоджену програму виконати, записуючи отримані результати.
- Використовувану команду виводу перетворити в коментар, а вивід результатів організувати на кожній вітці розгалуження з застосуванням складеної команди.
- Змінений текст програми записати у власну папку: під зміненим іменем (***Prizvyshche_lr3v31_2***)
- Відкомпілювати програму і виконати, записуючи отримані результати.

- Використовувану умовну команду перетворити в коментар, а обчислення змінної A реалізувати із використанням команди **?.**
- Змінений текст програми записати у власну папку: під зміненим іменем (***Prizvyshche_lr3v31_3***)
- Відкомпілювати програму і виконати, записуючи отримані результати.
- Оформити звіт про виконану роботу.

Теоретичні відомості

Розгалужені обчислювальні процеси в програмах на мові C++ реалізуються при допомозі умовної команди (команди розгалуження).

Команда розгалуження ***if*** має дві форми: повну та коротку.

Повна така:

if (<логічний вираз>) <команда1>; else <команда2>;

Дія команди. Обчислюється значення логічного виразу. Якщо це значення істинне, то виконується ***команда***1, у протилежному випадку – ***команда***2.

Команда 1 та ***команда 2*** можуть бути порожніми, простими або складеними.

Коротка команда розгалуження ***if*** має вигляд:

if (<логічний вираз>) <команда 1>;

Складена команда. Під час написання програми може виникнути потреба трактувати декілька команд як одну. Така команда називається складеною. Складена команда – це конструкція такого вигляду:

```
{
 <команда 1>;
 <команда N>;
 }
```

Перед закриваючою дужкою ";" ставити обов'язково. Після дужки символ ";" записувати не потрібно.

Логічні вирази та логічні операції.

Логічний вираз – це засіб записування умов, що задовольняють деякий критерій. Логічний вираз може набувати значення ***true*** (істинність) або ***false*** (хибність). Логічні вирази бувають прості та складені. Простий – це два числових чи символьних вирази, з'єднані символом відношення:

Таблиця 1 – Символи відношення

Символ	Значення
<	менше
<=	менше або рівне
==	рівне
>=	більше або рівне

>	більше
!=	не рівне

Таблиця 2 – Логічні операції

Операція	Значення
&&	логічне множення (and)
||	логічне додавання (or)
!	логічне заперечення (not)

Складні логічні вирази обчислюються "раціональним способом". Наприклад, якщо у виразі

(A<=B)&&(B<=C)

виявилось, що А більше В, то всі вирази, як і його перша частина (***A<=B***), приймають значення "хибно", тому друга частина (***B<=C)*** не обчислюється.

Результат логічної операції **1**, якщо істина і **0** у протилежному випадку.

Таблиця істинності логічних операцій наведена в таблиці 3.

Таблиця 3 – Таблиця істинності логічних операцій

E1	E2	E1&&E2	E1||E2	!E1
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Логічним виразом може бути ціле число. Якщо воно не дорівнює нулю, то значення логічного виразу – ***true***, якщо це число 0 – ***false***.

Приклад Нехай ***x = 9.*** Унаслідок виконання команд

```
if (x > 7)
                y = pow(x, 2);
                else y = sqrt(x);
if (x <= 5)
z = exp(x);
          else z = ++x;
```

отримаємо ***y = 81, z = 10, x = 10.***

Умова хибна, якщо вона дорівнює нулю, в інших випадках вона істинна. Це означає, що навіть від'ємні значення розглядаються як істинні. До того ж, умова, що перевіряється, повинна бути скалярною, тобто зводитися до простого значення, яке можливо перевірити на рівність нулю. Більш досвідчені програмісти скорочують команди типу:

if (вираз!=0) команда;

до наступної:

if (вираз) команда;

Обидва логічні вирази функціонально еквівалентні, тому що будь-яке ненульове значення розцінюється як істина

Команда ?. Команда **?** є аналогом команди розгалуження if. Загальний вигляд команди ? такий:

<логічний вираз> ? <команда або вираз1> : <команда або вираз2>

Дія команди. Обчислюється значення логічного виразу. Якщо воно істинне, то виконується команда 1 або обчислюється вираз 1, інакше – команда або вираз 2.

Приклад . Нехай ***c = 10***. Тоді після виконання команди

x = (c == 3) ? 2 * c : c - 2;

отримаємо ***x = 8***, оскільки ***c*** не дорівнює 3, і тому тут обчислюється значення виразу 2.

Наприклад, дану операцію зручно використати для знаходження найбільшого з двох чисел x і y:

max = (x > y) ? x : y;

Завдання

№ п/п	*Завдання*	X	Y
1	$A=\begin{cases} e^{xy}+D\ln x, & якщо\ \ x<y \\ \sqrt[3]{x+C}+\ln y, & якщо\ x\ge y \end{cases}$	0.128 7.521	2.936
2	$A=\begin{cases} \lg(Cx+y), & якщо\ \ x\le y \\ \sqrt[3]{x}+De^{y}, & якщо\ \ x>y \end{cases}$	2.587 5.394	3.962
3	$A=\begin{cases} tgC+xe^{y}, & якщо\ \ x>y \\ \lg y+Dx, & якщо\ \ x\le y \end{cases}$	-3.127 2.534	0.111
4	$A=\begin{cases} \lvert x+y\rvert+Ce^{x}, & якщо\ x\ge y \\ \ln(xy)+D, & якщо\ x<y \end{cases}$	-3.212 -0.555	-2.162
5	$A=\begin{cases} C\sqrt[3]{x+y}, & якщо\ x<y \\ De^{xy}, & якщо\ x\ge y \end{cases}$	0.275 4.598	0.637
6	$A=\begin{cases} \ln Cx+y^{2}, & \text{якщо}\ x>y \\ \sqrt[3]{Dy}+e^{x}, & \text{якщо}\ x\le y \end{cases}$	10.301 14.935	12.777
7	$A=\begin{cases} \ln\sqrt{x}+Cy, & якщо\ x\le y \\ tgxy+D\sqrt[3]{x}, & якщо\ x>y \end{cases}$	2.187 9.127	4.934
8	$A=\begin{cases} \cos 2x+3Cx^{2}, & якщо\ x>y \\ Dxy+\lg\lvert x+y\rvert, & якщо\ x\le y \end{cases}$	1.512 7.621	5.342
9	$A=\begin{cases} Ce^{xy}+\sqrt[3]{x}, & якщо\ x\ge y \\ \ln Dx, & якщо\ x<y \end{cases}$	0.793 7.581	5.469

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

10	$A=\begin{cases} C\lvert xy\rvert+\ln x, & якщо\ x>y \\ De^{\sqrt{x}}+\sqrt[3]{y}, & якщо\ x\le y \end{cases}$	1.275 3.984	2.564
11	$A=\begin{cases} e^{x}+C\ln(x^{2}), & якщо\ x>y \\ \sqrt[3]{xy}+Dy, & якщо\ x\le y \end{cases}$	2.537 6.758	5.163
12	$A=\begin{cases} C\ln xy, & якщо\ x<y \\ \sqrt[3]{(x+y)^{C}}, & якщо\ x\ge y \end{cases}$	2.089 7.597	3.869
13	$A=\begin{cases} D\ln x^{2}, & якщо\ x<y \\ C\sin y+e^{x}, & якщо\ x\ge y \end{cases}$	1.859 9.754	2.894
14	$A=\begin{cases} Ce^{xy}+\lg x, & якщо\ x<y \\ tg\ xy+Cy^{2}, & якщо\ x\ge y \end{cases}$	5.479 7.486	6.521
15	$A=\begin{cases} \sqrt{xy}+D\ln y, & якщо\ x>y \\ Cx^{2}+Dy, & якщо\ x\le y \end{cases}$	1.345 4.761	2.081
16	$A=\begin{cases} C^{xy}+\ln x, & якщо\ x<y \\ \cos x+\ln y, & якщо\ x\ge y \end{cases}$	0.354 2.658	1.993
17	$A=\begin{cases} xy+tgx, & якщо\ x\ge y \\ \sqrt[3]{x}+Dy^{2}, & якщо\ x<y \end{cases}$	5.812 6.111	5.992
18	$A=\begin{cases} \ln(xy)+yx, & якщо\ x\le y \\ \lvert x-y\rvert-Ce^{y}, & якщо\ x>y \end{cases}$	0.012 2.941	1.723
19	$A=\begin{cases} e^{y}+\sqrt{Cx+y}, & якщо\ x>y \\ \sqrt[3]{x}+Dy, & якщо\ x\le y \end{cases}$	1.643 3.759	2.007
20	$A=\begin{cases} Dtgx+xy, & якщо\ x\le y \\ e^{x}+\ln(xy), & якщо\ x>y \end{cases}$	0.043 3.879	1.369
21	$A=\begin{cases} Ce^{\sqrt{x}}+Dy, & якщо\ x\ge y \\ C\lvert xy\rvert+y, & якщо\ x<y \end{cases}$	1.648 4.987	3.922
22	$A=\begin{cases} x+\sqrt[3]{y}, & якщо\ x<y \\ \cos Cy-Dx, & якщо\ x\ge y \end{cases}$	2.438 7.113	3.289
23	$A=\begin{cases} xy-Cy\sqrt{y}, & якщо\ x\le y \\ e^{x}y^{2}+D, & якщо\ x>y \end{cases}$	4.378 8.523	7.132
24	$A=\begin{cases} Dx+Cy^{2}, & якщо\ x<y \\ \sqrt[3]{Dx+C}+\ln y, & якщо\ x\ge y \end{cases}$	0.231 4.687	3.277

25	$A=\begin{cases} Cxy-Dx\sqrt{y}, & якщо\ x\ge y \\ \sqrt[3]{xy}+Dy, & якщо\ x<y \end{cases}$	2.899 8.432	4.389
26	$A=\begin{cases} e^{xy}+Dx, & якщо\ x<y \\ Cx+\ln y, & якщо\ x\ge y \end{cases}$	3.098 7.888	5.872
27	$A=\begin{cases} e^{y}+\sqrt[3]{x}, & якщо\ x\le y \\ tgD+\ln y, & якщо\ x>y \end{cases}$	3.758 9.783	4.899
28	$A=\begin{cases} C\lvert x-y\rvert+\ln x, & якщо\ x\le y \\ e^{Cx}-\sqrt[3]{y}, & якщо\ x>y \end{cases}$	4.876 9.879	6.822
29	$A=\begin{cases} e^{Dx}+\ln y, & якщо\ x<y \\ \sqrt[3]{Cxy}+D\ln y, & якщо\ x\ge y \end{cases}$	2.863 7.982	4.921
30	$A=\begin{cases} D\ln x-e^{y}, & якщо\ x<y \\ \sqrt[3]{x+Cy}+\ln y, & якщо\ x\ge y \end{cases}$	2.954 8.693	7.772

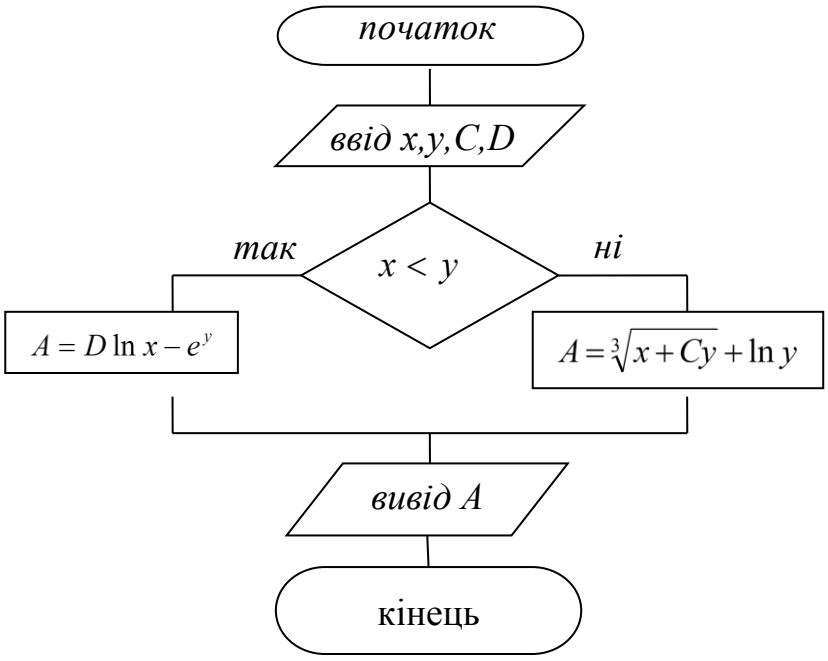
Приклад виконання завдання

Завдання:

Обчислити значення *A*:

Завдання	X	Y
$A=\begin{cases} D\ln x-e^{y}, & x<y \\ \sqrt[3]{x+Cy}+\ln y, & x\ge y \end{cases}$	2.954 8.693	7.772

Блок схема:



Програма1:

```
// Лабораторна робота №3 варіант 31 група СІ-11 Коломієць М.В.
#include <iostream>
#include <math.h>
using namespace std;

int main()
{
      system("color F0"); // Встановити білий фон і чорний текст
   float C=2.1 , D=0.56;
   float x,y,A;
   cout<<"x= "; cin>>x;
   cout<<"y= "; cin>>y;
   if(x<y)
        A = D * log(x) – exp(y);
          else
                A = pow(x + C * y , 1.0/3) + log(y);
   cout<<"A= "<<A<<endl;
   return 0;
}
```

Програма2:

```
// Лабораторна робота №3 варіант 31 група СІ-11 Коломієць М.В.
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
   system("color F0"); // Встановити білий фон і чорний текст
      float C=2.1 , D=0.56;
   float x,y,A;
   cout<<"x= "; cin>>x;
   cout<<"y= "; cin>>y;
   if(x<y)
        { A = D * log(x) – exp(y);
                cout<<"A= "<<A<<endl; }
          else
                { A = pow(x + C * y , 1.0/3) + log(y);
            cout<<"A= "<<A<<endl; }
   return 0;
}
```

Програма3:

```
// Лабораторна робота №3 варіант 31 група СІ-11 Коломієць М.В.
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
```

```
        system("color F0"); // Встановити білий фон і чорний текст
    float C=2.1 , D=0.56;
    float x,y,A;
    cout<<"x= "; cin>>x;
    cout<<"y= "; cin>>y;

     A = (x<y)? D * log(x) – exp(y) : pow(x + C * y , 1.0/3) + log(y);
    cout<<"A= "<<A<<endl;
    return 0;
}
```

Контрольні питання:

1. Основні стандартні математичні функції в С++.
2. Арифметичні вирази. Правила їх запису в С++.
3. Призначення та структура опису констант в С++.
4. Призначення та структура опису змінних в С++.
5. Команда присвоєння в С++. Синтаксис та порядок роботи.
6. Команда вводу в С++. Синтаксис та порядок роботи.
7. Команда виводу в С++. Синтаксис та порядок роботи.
8. Організація діалогу при вводі значень змінних в С++.
9. Послідовність компіляції програми в в інтегрованому середовищі Dev-С++.
10. Встановлення шрифту кирилиці в редакторі інтегрованого середовища Dev-С++.
11. Логічні вирази. Правила їх запису в С++. Істинність логічних операцій.
12. Умовна команда в С++. Синтаксис та порядок роботи.
13. Складена команда в С++. Синтаксис та порядок роботи.
14. Команда **?**. Синтаксис та порядок роботи.

Лабораторна робота №4
(2 год.)

Тема: Циклічний обчислювальний процес. Команда циклу з визначеною кількістю повторень ***for.***

Мета роботи: Оволодіти практичними навиками розробки та програмування обчислювального процесу циклічної структури. Засвоїти запис і використання команд операторів циклу з визначеною кількістю повторень ***for.*** Навчитись використовувати кому, як команду та вивід результатів у звичайному та науковому форматах.

За час виконання лабораторної роботи студент повинен освоїти:

- Розробку алгоритмів циклічної структури з використанням блоку модифікації.
- Синтаксис та застосування команди циклу з визначеною кількістю повторень.
- Сфери використання команди циклу з визначеною кількістю повторень.
- Запис команди циклу обчислення скінчених сум та добутків.
- Використання коми як команди.

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

- Організацію виводу результатів в різних форматах.

Завдання на лабораторну роботу: Скласти схему алгоритму та дві програми на С++ обчислення значення змінної **S** застосовуючи команду циклу з кроком (з визначеною кількістю повторень) у звичайному виді та з використанням коми, як команди.

Порядок виконання лабораторної роботи:

- Завантажити інтегроване середовище С++;
- Ввести текст програми (в першій стрічці обов'язково має бути коментар, в якому вказати номер роботи, групу та прізвище виконавця).
- Текст програми записати у власну папку: (***Prizvyshche_lr4v31_1***).
- Відкомпілювати програму, виправляючи при цьому можливі помилки.
- Відлагоджену програму виконати, записуючи отримані результати.
- Модифікувати створену програму, реалізувавши тіло циклу всередині заголовку команди циклу ***for*** із використанням коми, як команди. Вивід результатів передбачити у науковому форматі.
- Змінений текст програми записати у власну папку: під зміненим іменем (***Prizvyshche_lr4v31_2***)
- Відкомпілювати програму і виконати, записуючи отримані результати.
- Оформити звіт про виконану роботу.

Теоретичні відомості

Обчислювальний процес, що багаторазово повторюється при зміні деякої величини (параметр циклу) по наперед визначеному закону, називається циклічним обчислювальним процесом.

У мові С++ є три команди циклу:

- **for –** цикл з визначеною кількістю повторень (з лічильником);
- **while** – цикл з передумовою**;**
- **do – while** – цикл з післяумовою**.**

Команда ***for*** має вигляд

for (<вираз1>; <логічний вираз2>; <вираз3>) <команда1>;

Вираз1 призначений для підготовки циклу і виконується один раз. Переважно тут задають початкові значення змінних циклу (підготовляють цикл). У ***вираз2*** записують умову виходу із циклу. У ***вираз3*** – команди зміни параметрів циклу. ***Вираз1*** і ***вираз3*** або один із них у команді ***for*** можуть бути відсутні. У цьому випадку опускати символ ";" не можна. Наприклад, ***for(; i<10 ;) i++;.***

Якщо за допомогою одного із ***виразів*** необхідно виконати декілька дій, то використовують команду **"*кома*".**

Кома як команда. Кому, як команду, використовують тоді, коли необхідно інтерпретувати декілька виразів або команд як одне ціле. Вона має вигляд

вираз1, вираз2

або

команда 1, команда 2

Дія команди "кома". Послідовно обчислюються значення ***вираз1***

(виконується ***команда1***) та ***вираз2*** (***команда2)***.

Цю команду зручно використовувати у командах циклу, умовних командах тощо. Наприклад, ***if (k+=2, k<7) …***

Тут спочатку значення змінної ***k*** буде збільшено на ***2***, а потім це значення порівнюватиметься із числом ***7***. Результат команди - ***true,*** якщо значення змінної ***k*** менше, ніж ***7***, у протилежному випадку - ***false***.

Команда виходу з циклу *break*

Синтаксис : ***break***;

Дана команда перериває виконання команд циклу.

Коли команда ***break*** зустрічається всередині внутрішньої ***команди1*** (яка в такому випадку, як правило, є складеною), то здійснюється негайний вихід з циклу і перехід до виконання команди, що є наступною за командою циклу.

Команда продовження циклу continue

Синтаксис : ***continue***;

Команда ***continue*** передає управління на наступну ітерацію в командах циклу.

Роботу циклу for можна описати такими кроками:

1) Параметру циклу присвоюється початкове значення ***(вираз1).***
2) Обчислюється ***логічний вираз2.*** Якщо він має значення 0, тобто умова невірна, цикл припиняється.
3) Виконується внутрішня ***команда1*** циклу.
4) Параметр циклу змінює своє значення, що задає ***вираз3***.
5) Надалі перехід до пункту під номером 2.

Поява у будь-якому місці ***команди1*** команди ***continue*** призведе до негайного переходу до пункту 4.

Приклад 1. Суму цілих чисел з проміжку від ***1*** до ***15*** можна обчислити одним із способів:

```
int n = 1, S = 0;
for (; n < 16; n++) S += n;
for (int n = 1, S = 0; n < 16; n++) S+=n;
for (int n = 1, S = 0; n< 16; S+=n++);
for (int n = 1, S = 0; n< 16; S+=n, n++);
```

У результаті виконання команд змінній *S* буде присвоєно значення ***120.***

Зауважимо, що у способах ***2,3,4*** змінні **S** і **n** ініціалізовано (оголошено) безпосередньо у команді циклу ***for***.

Приклад 2. Кількість і добуток усіх парних цілих чисел із проміжку від ***4*** до ***11*** можна обчислити так:

```
int n, D, kil;
for (D = 1, kil = 0, n = 4; n <= 11; n += 2)
      { D*=n; kil++; }
```

<u>Варіанти завдань лабораторної роботи</u>

Обчислити значення **S**. Сталу **a** прийняти рівною **2.1**

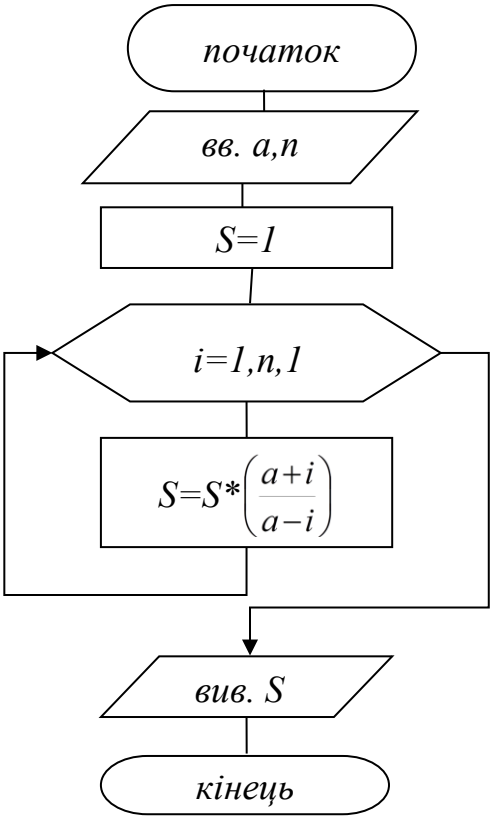
Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

№ п/п	Вираз для S	№ п/п	Вираз для S
1.	$\sum_{i=1}^{10}(ai+2)^2$	16.	$\prod_{i=1}^{4}(a+i)$
2.	$\sum_{i=1}^{20}\left(\frac{a}{i}\right)$	17.	$\prod_{i=1}^{5}(ai+2)$
3.	$\sum_{i=1}^{12}\left(\left(\frac{1}{2}\right)^2+a^i\right)$	18.	$\prod_{i=1}^{10}\left(\frac{a+i}{i^2}\right)$
4.	$\sum_{i=1}^{5}(2ai)$	19.	$\prod_{i=1}^{7}\left(\frac{i+2}{a+i}\right)$
5.	$\sum_{i=1}^{7}(i^2+ai)$	20.	$\prod_{i=1}^{8}(i^2+a)$
6.	$\sum_{i=1}^{14}\left(i+\frac{a^2}{i}\right)$	21.	$\prod_{i=1}^{9}(i^a)$
7.	$\sum_{i=1}^{9}(ai^3-3i)^2$	22.	$\prod_{i=1}^{11}(a^i)$
8.	$\sum_{i=1}^{14}(a^i+2i)$	23.	$\prod_{i=1}^{14}(i+ai^2)$
9.	$\sum_{i=1}^{19}(i+i^a)$	24.	$\prod_{i=1}^{19}\left(\frac{a+i}{ai}\right)$
10.	$\sum_{i=1}^{11}\left(\frac{i}{a+i}\right)$	25.	$\prod_{i=1}^{12}\left(\frac{ai}{a+i}\right)$
11.	$\sum_{i=1}^{4}\left(\frac{a^2+i^2}{2i}\right)$	26.	$\prod_{i=1}^{17}(a^2+i^2)$
12.	$\sum_{i=1}^{15}\left(\frac{a+i}{a+i^2}\right)$	27.	$\prod_{i=1}^{18}\left(\left[\frac{a}{i}\right]^2+ai\right)$
13.	$\sum_{i=1}^{21}\left(ai+\frac{i}{a}\right)$	28.	$\prod_{i=1}^{16}\left(\frac{a+i}{i^2+a}\right)$
14	$\sum_{i=1}^{18}(a^i+i^a)$	29.	$\prod_{i=1}^{22}\left(\frac{i^2}{a^i}\right)$
15.	$\sum_{i=1}^{17}\left(i^2+\frac{a^2}{i}\right)$	30.	$\prod_{i=1}^{20}\left(\frac{a+i}{a-i}\right)$

Приклад виконання завдання

Завдання: Обчислити значення S: $S = \prod_{i=1}^{20}\left(\frac{a+i}{a-i}\right)$

Блок схема:



Програма1:

```
// Лабораторна робота №4 варіант 31 група СІ-11 Коломієць М.В.
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
      system("color F0"); // Встановити білий фон і чорний текст
   float a=2.1 , S;
   int i , n;
   cout<<"n= "; cin>>n;
   S = 1;
   for (i = 1; i <= n; i++)
     S*= (a + i) / (a – i);
   cout<<"S= "<<S<<endl;
   return 0;
}
```

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

Зауваження Значення дійсних чисел (тип ***float***) можна виводити на екран у стандартному або науковому форматах. Якщо значення даного необхідно заокруглити до ***n*** значущих цифр, то перед командою виведення треба записати ***cout.precision(n).***

Для подання результатів у науковому форматі необхідно під'єднати файл заголовка ***iomanip.h*** і перед командою виведення записати

```
cout << setiosflags(ios::scientific);
```

Програма2:

```
#include <iostream>
#include <cmath>
#include <iomanip.h>
using namespace std;
int main()
{  system("color F0"); // Встановити білий фон і чорний текст
   float a=2.1 , S;
   int n;
   cout<<"n= "; cin>>n;
   S = 1;
   for (int i = 1; i <= n; S*= (a + i) / (a - i) , i++);
   cout << setiosflags(ios::scientific);
   cout<<"S= "<<S<<endl;
   return 0;
}
```

Контрольні питання:

1. Команда циклу з кроком в **C++**. Синтаксис та порядок роботи.
2. Сфери використання команди циклу з кроком.
3. Команда "кома". Синтаксис та порядок роботи
4. Основні стандартні математичні функції в **C++.**
5. Арифметичні вирази. Правила їх запису в **C++**.
6. Призначення та структура опису констант в **C++.**
7. Призначення та структура опису змінних в **C++.**
8. Команда присвоєння в **C++**. Синтаксис та порядок роботи.
9. Команда вводу в **C++.** Синтаксис та порядок роботи.
10. Команда виводу в **C++.** Синтаксис та порядок роботи.
11. Варіанти виводу дійсних чисел в **C++.**
12. Організація діалогу при вводі значень змінних в **C++.**
13. Логічні вирази. Правила їх запису в **C++**. Істинність логічних операцій.
14. Умовна команда в **C++.** Синтаксис та порядок роботи.
15. Складена команда в **C++.** Синтаксис та порядок роботи.
16. Команда **?**. Синтаксис та порядок роботи.

Лабораторна робота №5
(2 год.)

Тема: Циклічний обчислювальний процес. Команди циклу з передумовою та післяумовою.

Мета роботи: Оволодіти практичними навиками розробки та програмування обчислювального процесу циклічної структури. Засвоїти запис і використання команд циклу з передумовою та післяумовою. Навчитись організовувати циклічний обчислювальний процес табуляції функцій при допомозі всіх трьох команд циклу.

За час виконання лабораторної роботи студент повинен освоїти:

- Розробку алгоритмів циклічної структури з передумовою та післяумовою.
- Синтаксис та застосування команди циклу з передумовою.
- Синтаксис та застосування команди циклу з післяумовою.
- Відмінності та сфери використання команди циклу з передумовою та післяумовою.
- Запис складних виразів з використанням формул представлення одних функцій через інші.
- Організацію виводу результатів в табличній формі.

1) ***Завдання на лабораторну роботу:*** Скласти три схеми алгоритму та три програми на ***C++*** обчислення значення функції на проміжку від ***xp*** до ***xk*** з кроком ***h,*** застосовуючи команду циклу з передумовою, команду циклу з післяумовою та команду циклу з визначеною кількістю повторень ***for***. Вивід результатів передбачити в виді таблиці з заголовками колонок "***аргумент***", "***функція***".

Порядок виконання лабораторної роботи:

- Використовуючи відповідний ярлик, викликати інтегроване середовище ***C++:***
- Ввести текст програми, в якій циклічний обчислювальний процес реалізований із використанням команди циклу з передумовою (в першій стрічці обов'язково має бути коментар, в якому вказати номер роботи, групу та прізвище виконавця). Ввід вхідних даних має бути організований з використанням комбінації команд виводу і вводу, а вивід результатів – у науковому форматі.
- Текст програми записати у власну папку: (***Prizvyshche_lr5v31_1***).
- Відкомпілювати програму, виправляючи при цьому можливі помилки.
- Відлагоджену програму виконати, записуючи отримані результати.
- Модифікувати створену програму, реалізувавши циклічний обчислювальний процес із використанням команди циклу з післяумовою.
- Змінений текст програми записати у власну папку під зміненим іменем (***Prizvyshche_lr5v31_2***)
- Відкомпілювати програму, виправляючи при цьому можливі помилки.
- Відлагоджену програму виконати, записуючи отримані результати.

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

- Модифікувати одну із створених програм, реалізувавши циклічний обчислювальний процес із використанням команди циклу з визначеною кількістю повторень ***for***.
- Змінений текст програми записати у власну папку під зміненим іменем (***Prizvyshche_lr5v31_3***)
- Відкомпілювати програму, виправляючи при цьому можливі помилки.
- Відлагоджену програму виконати, записуючи отримані результати.
- Оформити звіт про виконану роботу.

Теоретичні відомості

Команда циклу з передумовою (***while***) має вигляд:

***while* (<*логічний* вираз>) <*команда1*>;**

Дія команди.

1) Обчислюється значення ***логічного виразу***. Якщо воно істинне, то переходимо до пункту 2), якщо хибне – до пункту 3).
2) Виконується ***команда1*** і відбувається перехід до пункту 1).
3) Відбувається перехід до наступної після ***while*** команди.

Команда1 може бути порожньою, простою або складеною.

Для того, щоб відбувся вихід із циклу, необхідно змінювати значення параметра циклу у ***команді1***. Параметр циклу – це змінна, що входить у ***логічний вираз***.

Команда ***while*** може бути виконана нуль або більше раз.

Приклад

Нехай ***x = 4, s = 0, d = 1***. Після виконання команд

```
while (x <= 8)  { s += x; x++;}
while (x> 10) d*=x;
```

s = 0 + 4 + 5 + 6 + 7 + 8 = 30, x = 9,

а змінна ***d*** свого значення (**1**) не змінить, оскільки значення виразу ***x > 10*** хибне, і тому команда ***d *= x*** у циклі **while** виконуватись не буде.

Задача **(про суму цифр натурального числа).** Для заданого натурального числа *n* підрахувати суму його цифр

```
#include <iostream>
 using namespace std;

void main()
{
  int S = 0, n, m;
  cout << "Введіть число "; cin >> n;
  m = n;
  while (m > 0)
        {
          S+=m % 10;
          m/=10;
        }
  cout << "Сума цифр числа " << n << " дорівнює " << S;
}
```

Команда циклу з післяумовою ***do while*** має вигляд:

do <***команда1***>; ***while (<логічний вираз>);***

Дія команди.

1) Виконується ***команда1*** і обчислюється значення ***логічного виразу***.
2) Якщо значення ***логічного виразу*** істинне, то див. пункт 1), якщо значення ***логічного виразу*** хибне – відбувається перехід до наступної після ***do while*** команди.

Команда1 у циклі ***do while,*** на відміну від циклу ***while,*** буде виконуватись хоча б один раз завжди.

Приклад. Нехай цілі змінні ***x, y*** мають значення ***x = 5, y = 0.*** У результаті виконання команд

```
do {
        y += x; z = 2 * x; x -= 2;
  }
 while (x > 1);
```

змінні ***x, y, z*** набудуть таких значень: ***y = 0 + 5 + 3 = 8, z = 6, x = 1.***

Задача **(про розклад числа на прості множники).**

Розкласти на прості множники задане натуральне число. Кількість виведень кожного множника дорівнює його кратності у розкладі.

```
#include <iostream>
using namespace std;

void main()
{
 int i, n;
 cout << "Введіть число "; cin >> n;
cout << "\n n = 1";
for(i = 2; i <= n; i++)
{
  if (n % i) continue;
  do
{
  cout << " * " << i;
    n /= i;
  }
     while (!(n%i));
    }
 }
```

Варіанти завдань лабораторної роботи

№ п/п	функція	a	b	xp	xk	h
1	$y = \lg\left\|\arcsin(0.01 \times ax^2)\right\| + \sqrt[3]{\left\|\mathrm{tg}(bx)\right\|} + e^{3.7}$	0.12	0.11	1	3	0.2
2	$y = e^{2x}\,\mathrm{tg}^2(ax) + \lg(\arccos(0.1 \times bx))^2 + x^{3.1}$	0.99	0.32	2	3	0.1
3	$y = \lg\left\|\mathrm{tg}(ax)\right\| + \dfrac{\sqrt[7]{bx^2}}{\arcsin x} - e^{(a+x)}$	0.47	0.56	0.07	0.49	0.07
4	$y = a(\arccos x)^2 - e^{2.8}\,\mathrm{tg}(bx) + \lg\sqrt[5]{x}$	0.28	0.76	0.46	0.66	0.04

5	$y = \arcsin x \times \operatorname{tg}^2(ax) - \lg(2x) + \frac{e^{bx}}{\sqrt[3]{x}}$	0.47	0.98	0.12	0.72	0.06
6	$y = bx^{7.1} \lg(ax) + \frac{e^{ab} \operatorname{tg}^2(ax)}{\arccos(0.05 \times bx)}$	0.69	0.46	π/2	π	π/10
7	$y = e^x \frac{\lg(x^2 + b)\arcsin(b)}{\operatorname{tg}(3x) + \sqrt[3]{xa}}$	0.43	0.88	2	8	0.5
8	$y = \lg\lvert\arccos(ax)\rvert + e^{2.5} \operatorname{tg}(x^2 + b)^{4.1}$	0.19	0.33	0.22	0.82	0.06
9	$y = \frac{\sqrt{x}}{(x+b)^9} - \frac{\lg(xa)}{\arccos x} + \operatorname{tg} b + e^{(2x-a)}$	0.29	0.99	0.17	0.27	0.01
10	$y = \frac{e^{9.2} \arcsin b}{\operatorname{tg}(ax) + \sqrt[5]{x+a}} + \lg(2x^2)$	0.37	0.46	1.5	7.5	0.5
11	$y = \frac{\operatorname{tg}(xa)}{\arcsin(0.05 \times ax)} + e^{3.7} \lg(b^2) - \sqrt[3]{bx}$	0.58	0.37	2.5	8	0.5
12	$y = \arccos x + \operatorname{tg}(x^2 + b) - \lg\sqrt{x} + (ax)^{7.3} e^{9.4}$	0.26	0.11	0.08	0.8	0.08
13	$y = \frac{e^{7.9}}{\arccos(x-a)} + \lg(x^2 + b^2)\sqrt[3]{bx} \times \operatorname{tg}(ax)$	0.65	0.27	1.01	1.21	0.02
14	$y = \frac{tg(xa)}{e^{1.2} \arccos x} + \frac{\lg(bx+a)}{\sqrt[5]{x + ba^2}}$	0.98	0.98	0.1	0.8	0.07
15	$y = \frac{a\sqrt[3]{x} e^{3.7} + \operatorname{tg}(2x^2) \lg(5x)}{\arcsin(x + 0.2b)}$	0.57	0.47	0.02	0.2	0.02
16	$y = \frac{e^{4.7\sqrt[3]{ax + b^2}} - \lg(bx)}{\operatorname{tg}(bx) + \arccos(b)}$	0.37	0.59	1.3	4	0.3
17	$y = \sqrt[3]{a^2 + x} - e^{3.6} \operatorname{tg}(a + x)\arcsin(b^2) + \lg(\sqrt{x} + b)$	0.87	0.28	1.42	9.42	0.25
18	$y = \lg(\sqrt{x} + a)\operatorname{tg}(x^2 - b) + \frac{e^{1.9} \arcsin(a+x)\sqrt[3]{xa}}{\operatorname{tg} b}$	0.05	0.49	0.07	0.7	0.07
19	$y = \sqrt[7]{x^2 + \sqrt{x}} e^{1.5} \lg x - \arccos a + \operatorname{tg}(2b)$	0.97	0.54	3	15	2
20	$y = \lg(b^2 + \sqrt{2x})\operatorname{tg}(x + a)e^{7.2} + \arcsin b - \sqrt[9]{x}$	0.98	0.25	2.75	6.75	0.5
21	$y = \operatorname{tg}^2 x + \lg\sqrt{x} \times \arccos(bx) + e^{1.3}\sqrt[3]{(x+a)^2}$	0.98	0.1	1.04	5.04	0.5
22	$y = \frac{\lg(\sqrt{x} + bx^2)}{\sqrt[9]{a+x}} e^{3.9} - \operatorname{tg}(a+b) + \arccos b$	0.87	0.43	2.54	6.54	0.4
23	$y = \arcsin(xb) \times \lg\lvert xa^2 - \sqrt{ax}\rvert + e^{0.9} \times \sqrt[3]{x} \times \operatorname{tg}(x + ba)$	0.37	0.75	0.04	0.44	0.04

24	$y=\frac{e^{0.8}\,\mathrm{tg}(xa)}{\arccos(ax)}+\frac{\lg\left(ax^2+b\sqrt{x}\right)}{\sqrt[3]{4x^2}}$	0.04	0.98	1	7	0.6
25	$y=\lg\left(\sqrt[3]{x}+b\right)+e^{1.2}\,\mathrm{tg}\left(a+b^2\right)+\arcsin(bx)$	0.58	0.56	0.21	0.41	0.02
26	$y=\mathrm{tg}^2\sqrt{b}+\lg\left(x^{1.2}\right)e^{3.8}+\arccos(x+0.2a)$	0.37	0.37	0.22	0.82	0.06
27	$y=\arcsin(a)+\lg\left(\sqrt{ax}+b^2\right)\mathrm{tg}(bx)+e^{0.3\sqrt[3]{x^2}}$	0.36	0.97	5	15	1
28	$y=\frac{\mathrm{tg}\left(x^2+\sqrt{b}\right)}{\sqrt[3]{ax}}+\lg(x+2b)\arcsin(bx)+e^{0.4}$	0.95	0.09	0.54	0.84	0.03
29	$y=\sqrt[5]{bx^2}+e^{3.7}\lg\left(\sqrt{x}+a^2\right)\mathrm{tg}\,x-\arccos(2x)$	0.22	0.38	0.19	0.49	0.03
30	$y=e^{0.4}\arcsin x-\lg\left\|\mathrm{tg}\left(x+\sqrt{b}\right)\right\|+\sqrt[7]{x^2+a}$	0.88	0.77	0.24	0.64	0.04

Приклад виконання завдання

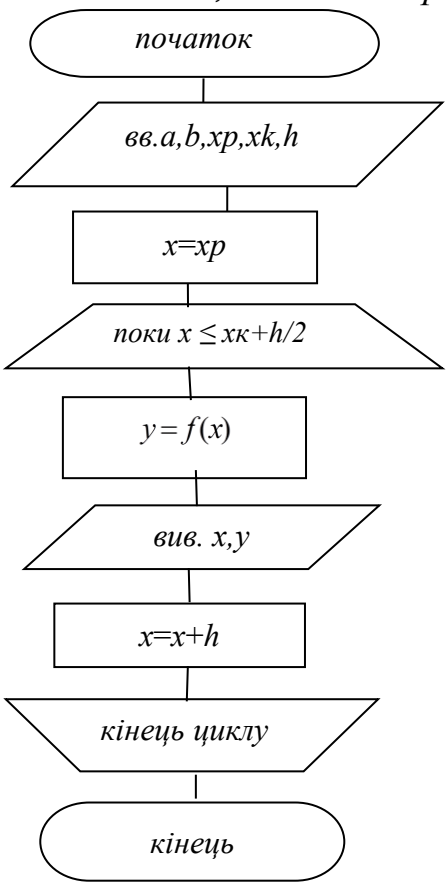
Завдання:

функція	a	b	xp	xk	h
$y=e^{0.4}\arcsin x-\lg\left\|\mathrm{tg}\left(x+\sqrt{b}\right)\right\|+\sqrt[7]{x^2+a}$	0.88	0.77	0.24	0.64	0.04

Програма з командою while:

```
#include <iostream>
#include <cmath>
#include <iomanip.h>

int main()
{
   system("color F0"); // Встановити білий фон і чорний текст
   setlocale(0, "ukr");
   float a,b,xp,xk,h,x,y;
   cout << "a=  "; cin >> a;
   cout << "b=  "; cin >> b;
   cout << "xp= "; cin >> xp;
   cout << "xk= "; cin >> xk;
   cout << "h=  "; cin >> h;
   x = xp;
   cout << "аргумент" << "\t" << "функція " << endl;
   cout << setiosflags(ios::scientific);
   while(x <= xk + h/2)
   {
    y = exp(0.4) * asin(x) - log10(fabs(tan(x + sqrt(b)))) +
        pow(x*x + a,1.0/7);
    cout << x << "\t" << "\t" << y << endl;
    x += h;
   }
   return 0;
}
```

Блок схема для програми з командою while:



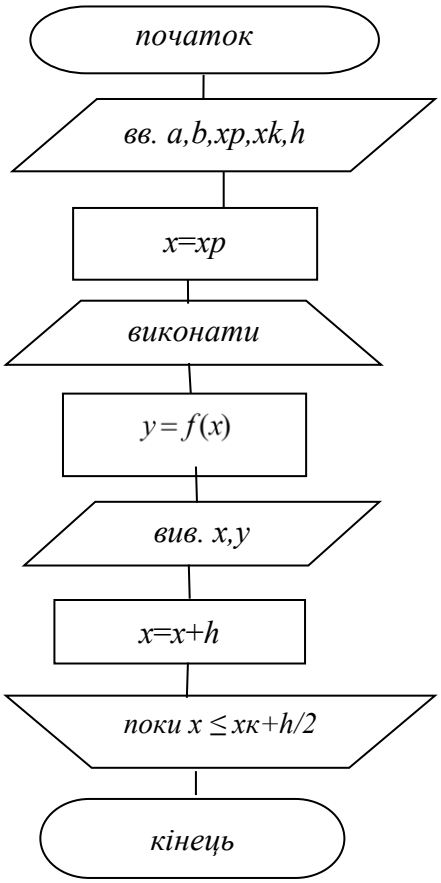
Зауваження Ураховуючи специфіку комп'ютерної арифметики, щоб не втратити останнього значення функції, у будові циклів ***while, do while***, ***for*** до останнього значення ***xk*** додаємо деяке зміщення, у цьому випадку ***h/2.***

Програма з командою do while:

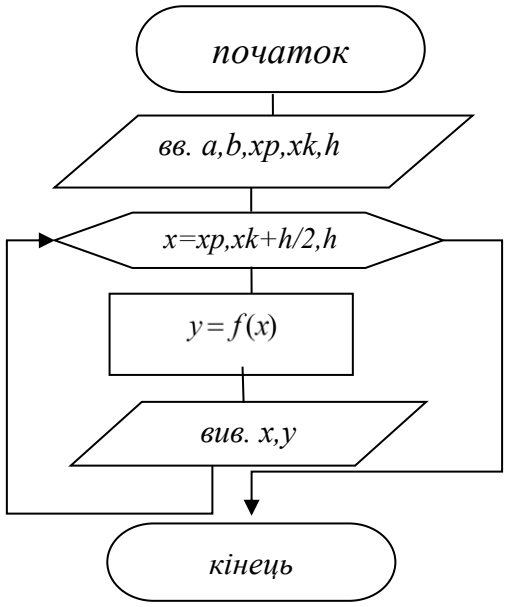
```
#include <iostream>
#include <cmath>
#include <iomanip.h>

int main()
{
      system("color F0"); // Встановити білий фон і чорний текст
      setlocale(0, "ukr");
      float a,b,xp,xk,h,x,y;
   cout << "a=  ";  cin >> a;
   cout << "b=  ";  cin >> b;
   cout << "xp= ";  cin >> xp;
   cout << "xk= ";  cin >> xk;
   cout << "h=  ";  cin >> h;
   x = xp;
   cout << "аргумент" << "\t" << "функція " << endl;
   cout << setiosflags(ios::scientific);
   do
   {
    y = exp(0.4) * asin(x) - log10(fabs(tan(x + sqrt(b)))) +
```

```
        pow(x*x + a,1.0/7);
     cout << x << "\t" << "\t" << y << endl;
     x += h;
    }
    while(x <= xk + h/2)
    return 0;
}
```

Блок схема для програми з командою do while:



Блок схема для програми з командою for:



Програма з командою for:

```
#include <iostream>
#include <cmath>
#include <iomanip.h>

int main()
{
   system("color F0"); // Встановити білий фон і чорний текст
   setlocale(0, "ukr");
   float a,b,xp,xk,h,x,y;
   cout << "a=  "; cin >> a;
   cout << "b=  "; cin >> b;
   cout << "xp= "; cin >> xp;
   cout << "xk= "; cin >> xk;
   cout << "h=  "; cin >> h;
   cout << "аргумент" << "\t" << "функція " << endl;
   cout << setiosflags(ios::scientific);
   for(x = xp; x<= xk + h/2; x+=h)
   {
    y = exp(0.4) * asin(x) - log10(fabs(tan(x + sqrt(b)))) +
        pow(x*x + a,1.0/7);
    cout << x << "\t" << "\t" << y << endl;
   }
   return 0;

}
```

Контрольні питання:

1. Команда циклу з передумовою в **C++**. Синтаксис та порядок роботи.
2. Команда циклу з післяумовою в **C++.** Синтаксис та порядок роботи.
3. Відмінності та сфери використання команд циклу з передумовою та післяумовою.
4. Організація діалогу при вводі значень змінних в **C++.**
5. Логічні вирази. Правила їх запису в **C++**. Істинність логічних операцій.

6. Умовна команда в **C++**. Синтаксис та порядок роботи.
7. Складена команда в **C++.** Синтаксис та порядок роботи.
8. Команда **?**. Синтаксис та порядок роботи.
9. Послідовність компіляції програми в в інтегрованому середовищі Dev-C++.
10. Встановлення шрифту кирилиці в редакторі інтегрованого середовища Dev-C++.
11. Логічні вирази. Правила їх запису в C++. Істинність логічних операцій.
12. Основні стандартні математичні функції в C++.

Вправи

Визначте результати виконання таких команд (усно):

а) a = 5; for (i = 1; i <= 2; I++) a = a * i - 2; a++; (відповідь: a = 5);
б) a = 1; for (i = 1; i <= 3; i++) {a = a + i; a -= 1;}
в) a = 0; for (i = 1; i <= 4; i++) a += i; a += 2;
г) p = 1; for (b = 8; b >= 5; p += b—); p++;
д) s = 0; for (n = 7; n >= 4; n--) {s += n; s++;}
е) a = 4; for (i = 1; i <= 2; i++) a =a * i - 1; a += 2;
є) a = 1; for (i = 1; i <= 3; i++) {a = a + 2 * i; a -= 2;}
ж) a = -1; for (i = 1; i <= 4; i++) a = 2 * a + i; a += 2;
з) p = 30; for (b = 7; b >= 4; b--) p -= b; p += 5;
і) s = 0; for (n = 6; n >= 3; n--) {s = s + 2 * n; s-- ;}.

Спеціальності: "Комп'ютерні науки", "Системний аналіз та наука про дані" "Інформаційні системи і технології", "Комп'ютерна інженерія"

НАВЧАЛЬНЕ ВИДАННЯ

МЕТОДИЧНІ ВКАЗІВКИ
для виконання лабораторних робіт
з дисципліни "ПРОГРАМУВАННЯ"
Частина 1

для студентів спеціальностей

F3 Комп'ютерні науки,
F4 Системний аналіз та наука про дані,
F6 Інформаційні системи і технології,
F7 Комп'ютерна інженерія.

Укладачі: к.т.н., доцент Гладьо Ю.Б.,
старший викладач Семенишин Г.М.,
к.т.н. старший викладач Гладьо С.В.,
к.т.н., доцент Гащин Н.Б.

Підписано до друку _____ Формат 60x84 1/16. Ум. др. арк. 1.
Друк лазерний. Замовлення № _____. Наклад 100 пр.
Віддруковано у видавництві ТНТУ.