

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Київський національний університет будівництва і архітектури

## **КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ТА ПРОГРАМУВАННЯ**

Методичні вказівки  
до виконання лабораторних робіт  
для здобувачів першого (бакалаврського) рівня вищої освіти  
за спеціальностями 174 «Автоматизація, комп'ютерно-інтегровані  
технології та робототехніка» та 141 «Електроенергетика, електротехніка  
та електромеханіка»

Київ 2025

УДК 004.021

К63

Укладачі: О.В. Бондарчук, канд. техн. наук, доцент;  
В.Ю. Луценко, доцент

Рецензент С.В. Іносов, канд. техн. наук, доцент

Відповідальний за випуск А.В. Заприво́да, канд. техн. наук,  
доцент

*Затверджено на засіданні кафедри автоматизації  
технологічних процесів, протокол №5 від 2 грудня 2024 року.*

**Комп'ютерні** технології та програмування : методичні вказівки  
до лабораторних робіт / уклад. : О.В. Бондарчук, В.Ю. Луценко. –  
Київ : КНУБА, 2025. – 40 с.

Розглянуто загальні методичні вказівки по проведенню  
лабораторних робіт. Значну увагу приділено прикладам розробки  
програм на мові програмування C++.

Призначено для здобувачів першого (бакалаврського) рівня  
вищої освіти за спеціальностями 174 «Автоматизація, комп'ютерно-  
інтегровані технології та робототехніка» та 141 «Електроенергетика,  
електротехніка та електромеханіка».

© КНУБА, 2025

## ЗМІСТ

ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	4
Лабораторна робота №1.....	8
Теоретичні відомості.....	8
Зміст лабораторної роботи.....	9
Контрольні запитання.....	12
Лабораторна робота №2.....	12
Зміст лабораторної роботи.....	12
Контрольні запитання.....	16
Лабораторна робота №3.....	17
Зміст лабораторної роботи.....	17
Контрольні запитання.....	23
Лабораторна робота №4.....	23
Зміст лабораторної роботи.....	23
Контрольні запитання.....	27
Лабораторна робота №5.....	27
Зміст лабораторної роботи.....	27
Контрольні запитання.....	31
Лабораторна робота №6.....	32
Зміст лабораторної роботи.....	32
Контрольні запитання.....	33
Лабораторна робота №7.....	34
Зміст лабораторної роботи.....	34
Контрольні запитання.....	37
Лабораторна робота №8.....	37
Зміст лабораторної роботи.....	37
Контрольні запитання.....	39
СПИСОК ЛІТЕРАТУРИ.....	39

## **ЗАГАЛЬНІ ПОЛОЖЕННЯ**

**Метою** виконання лабораторних робіт є закріплення набутих під час вивчення дисципліни «Комп'ютерні технології та програмування» теоретичних знань.

Завдання виконання лабораторних робіт полягає в закріпленні основних теоретичних питань, які стосуються головних принципів роботи в програмному середовищі Visual Studio, а також у створенні студентами програм на мові програмування C++.

Мову C++ можна віднести до класу мов програмування проміжного (або середнього) рівня. Це означає, що застосування мови C++ дозволяє створювати як високорівневі програми, так і низькорівневі бібліотеки, які забезпечують взаємодію з апаратними засобами ЕОМ. Для багатьох програмістів мова C++ є оптимальною комбінацією, оскільки, будучи мовою високого рівня, дозволяє створювати складні програми, забезпечуючи при цьому можливість досягнення високої продуктивності саме за рахунок контролю над використанням ресурсів та їх доступністю.

### **1. ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ ПО ПРОВЕДЕННЮ ЛАБОРАТОРНИХ РОБІТ**

Під час виконання лабораторних робіт студентам необхідно дотримуватися правил техніки безпеки та правил внутрішнього розпорядку університету. Перед початком роботи потрібно впевнитись, що апаратура не пошкоджена і викладач дозволив розпочати роботу. Під час роботи на комп'ютері необхідно дотримуватись оптимальної віддалі до екрана монітора (60 – 70 см), допустима відстань – не менше 50 см.

В кожній лабораторній роботі наведені приклади складання програм для задач аналогічного типу. Графічні схеми алгоритмів програм наведені лише там, де це було необхідно для пояснення принципів програмування.

Звіти лабораторних робіт повинні виконуватись кожним студентом самостійно. Одержаний результат лабораторної роботи необхідно показати викладачу.

До звіту входять: мета роботи, умова задачі відповідно до варіанта, код програми, результати роботи програми, висновок (пояснення до програми).

## 2. ПІДГОТОВКА ТА ВИКОНАННЯ ПРОГРАМ

### *Консольний режим роботи середовища Visual Studio C++ 19.0*

Програма, яка створюється в середовищі Visual Studio C++, завжди оформляється у вигляді окремого проєкту. Проєкт (**project**) – це набір взаємопов’язаних вихідних файлів, призначених для вирішення певного завдання. В результаті компіляції і компоунання цих файлів можна отримати готову до виконання програму. У проєкт входять як файли, що створюються програмістом безпосередньо, так і файли, які автоматично генеруються та редагуються самим середовищем програмування.

Для створення нового проєкту необхідно (рис.1):

- вибрати в меню **File** пункт **Create>Project**, щоб відкрити діалогове вікно «Create a new project»;

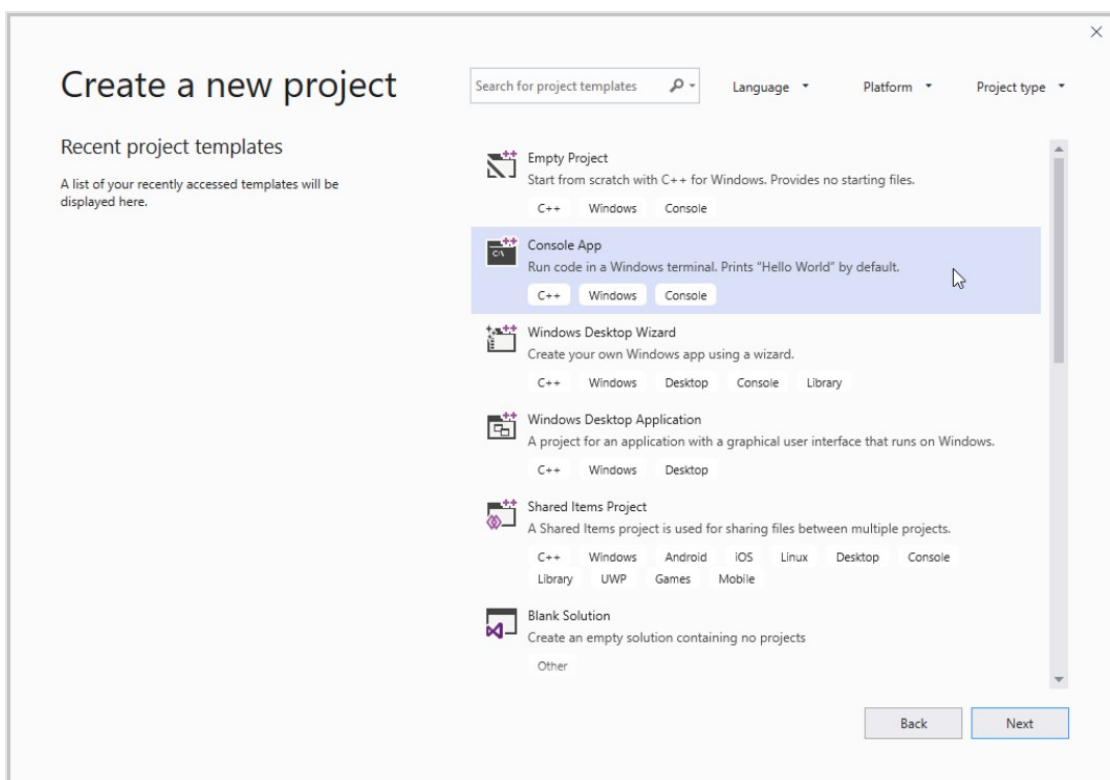


Рис. 1. Діалогове вікно створення нового проєкту

- у вікні обрати шаблон **Console App** з тегами *C++*, *Windows*, *Console* та натиснути кнопку **Next**;
- у діалоговому вікні **Configure your new project** (рис. 2) у полі **Project Name** ввести ім’я проєкту;

- у полі **Location** ввести ім'я каталогу, в якому буде розміщено проєкт і повний шлях до нього.
- Для створення проєкту натиснути кнопку **Create**.

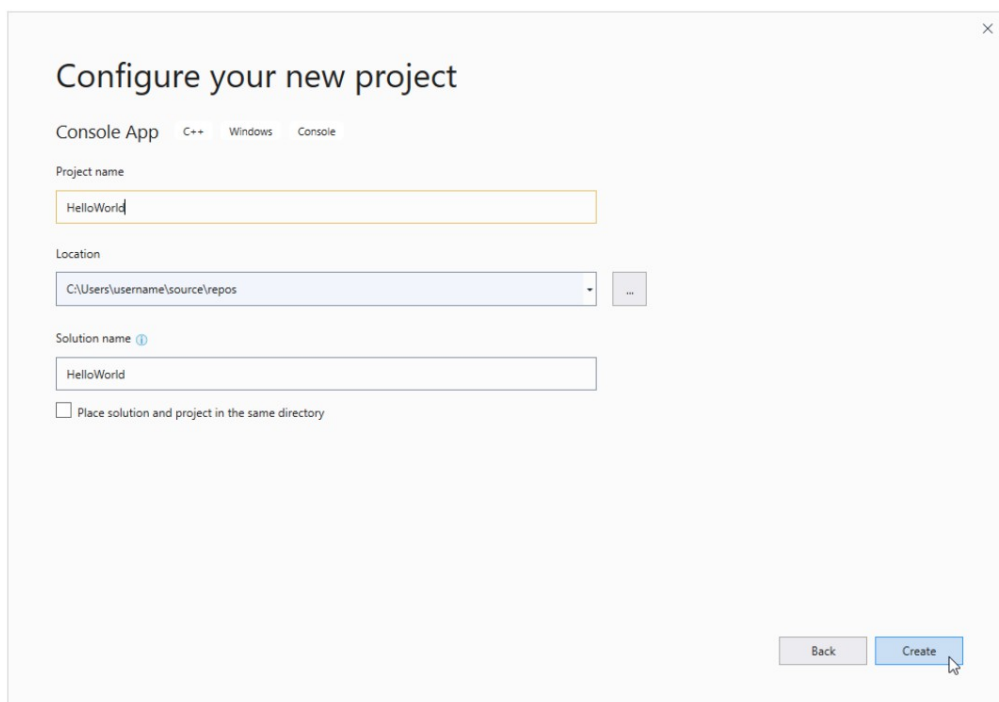


Рис. 2. Діалогове вікно початкових налаштувань проєкту

Для додавання в проєкт вже існуючого файлу з текстом програми необхідно:

- скопіювати існуючий файл (розширення .cpp) в робочу папку проєкту;
- у вікні **Workspace**, закладка **FileView**, клацнути правою кнопкою миші на **теці Source Files**;
- у діалоговому вікні **Insert Files...** вибрати новий файл і клацнути мишею по кнопці **OK**;

Visual Studio створює і можна приступати до додавання або редагування вихідного коду. За замовченням шаблон консольного додатка представляє вихідний код, як показано на рис. 3.

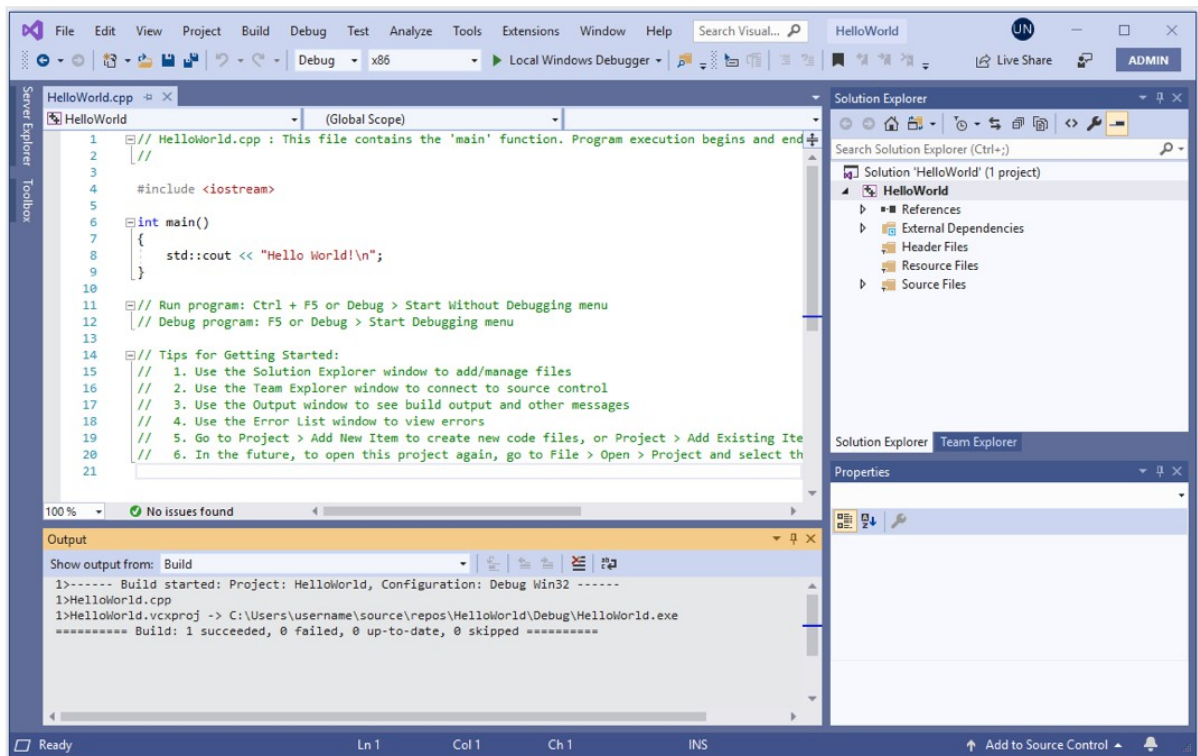


Рис. 3. Діалогове вікно для написання та редагування програмного коду

В папці проекту, як правило, розміщено п'ять файлів і одна вкладена папка. Структура проекту відображається у вікні «Оглядач проекту». Файли мають наступне призначення:

- файл з розширенням `dsw` (наприклад, `mylab1.dsw`) – файл проекту, який об'єднує всі вхідні в проєкт файли;
- файл з розширенням `dsp` (наприклад, `mylab1.dsp`) призначений для побудови окремого проєкту або підпроєкту;
- файл з розширенням `opt` (наприклад, `mylab1.opt`) містить всі налаштування даного проєкту;
- файл з розширенням `ncb` (наприклад, `lab1.ncb`) – службовий файл;
- файл з розширенням `cpp` (наприклад, `lab1.cpp`) файл тексту програми.

Результати компіляції виводяться у вікно **Output**. **Build** (F7) – компонування проєкту. Компілюються всі файли, в яких відбулись зміни з моменту останнього компонування. Якщо компонування пройшло без помилок, то середовище програмування створить виконуваний файл з розширенням `exe`, який можна буде запустити на виконання. **Execute** (Ctrl+F5) – виконання виконуваного файлу, створеного в результаті компонування проєкту. Для файлів, у які були внесені зміни, виконується

перекомпіляція і перекомпонування. Якщо в процесі компіляції було виявлено синтаксичні помилки, то виводиться відповідне повідомлення. У цьому випадку необхідно послідовно виправляти помилки і компілювати проєкт знову. Якщо синтаксичних помилок немає, але результат виконання програми неправильний, необхідно шукати логічні помилки. Для цього слід використовувати вбудований в систему відладчик. Після закінчення роботи проєкт можна закрити, вибравши **File – Close Workspace**, або закрити додаток. Для відкриття збереженого раніше проєкту необхідно вибрати **File – Open Workspace...** У діалоговому вікні вибрати папку проєкту та відкрити файл з розширенням dsw.

## Лабораторна робота №1

### АРИФМЕТИЧНІ ОПЕРАЦІЇ

**Мета роботи:** навчитися розробляти алгоритми програм та програми, які містять арифметичні оператори, та познайомитися із складом бібліотеки математичних функцій `cmath`.

#### Теоретичні відомості

Функції для обчислення математичних виразів знаходяться в бібліотеці `math.lib`.

Для підключення бібліотеки `math.lib` в заголовку файлу потрібно додати команду

```
#include <cmath>
```

Опис основних функцій, що складають зміст цієї бібліотеки, наведено в табл. 1.

*Таблиця 1*

#### Математичні функції (бібліотека `math.lib`)

Назва функції	Результат дії функції
<code>abs(X)</code>	модуль – повертає модуль числа $X$
<code>acos(X)</code>	арккосинус – повертає арккосинус числа $X$ в радіанах
<code>asin(X)</code>	арксинус – повертає арксинус числа $X$ в радіанах
<code>atan(X)</code>	арктангенс – повертає арктангенс числа $X$ в радіанах
<code>atan2(X,Y)</code>	арктангенс з двома параметрами – повертає арктангенс для заданих координат $X$ та $Y$ радіанах від $-\pi$ до $\pi$ за виключенням $\pi$ .

$\text{ceil}(X)$	округлення числа $X$ до найближчого більшого цілого – повертає число, округлене до найближчого більшого цілого
$\text{cos}(X)$	косинус – повертає косинус кута $X$
$\text{cosh}(X)$	гіперболічний косинус – повертає гіперболічний косинус числа $X$
$\text{exp}(X)$	експонента – повертає експоненту заданого числа $X$
$\text{fabs}(X)$	абсолютна величина числа $X$ – повертає модуль числа $X$ з плаваючою крапкою
$\text{floor}(X)$	округлення $X$ до найближчого меншого цілого – повертає число, округлене до найближчого меншого цілого
$\text{fmod}(X, Y)$	залишок від ділення – повертає залишок від ділення $X/Y$ для чисел з плаваючою крапкою
$\text{frexp}(X)$	розбиває число $X$ з плаваючою крапкою на мантису й показник степені
$\text{log}(X)$	натуральний логарифм числа $X$ – повертає натуральний логарифм числа $X$
$\text{log10}(X)$	логарифм по основі 10 числа $X$ – повертає десятинний логарифм числа $X$
$\text{modf}(X, P)$	виділяє цілу та дробову частини (з урахуванням знака) із числа з плаваючою крапкою $X$ (дробова частина буде знаходитися в змінній $P$ )
$\text{pow}(X, Y)$	повертає результат піднесення $X$ в ступінь $Y$ , $X^Y$
$\text{sin}(X)$	синус – повертає синус кута $X$
$\text{sinh}(X)$	гіперболічний синус – повертає гіперболічний синус числа $X$
$\text{sqrt}(X)$	квадратний корінь – повертає значення квадратного кореня з числа $X$
$\text{tan}(X)$	тангенс – повертає тангенс кута $X$
$\text{tanh}(X)$	гіперболічний тангенс – повертає гіперболічний тангенс числа $X$

### Зміст лабораторної роботи

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти і відкоригувати програму рішення задачі.

3. Проаналізувати роботу різних математичних функцій.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

### Приклад розробки програм

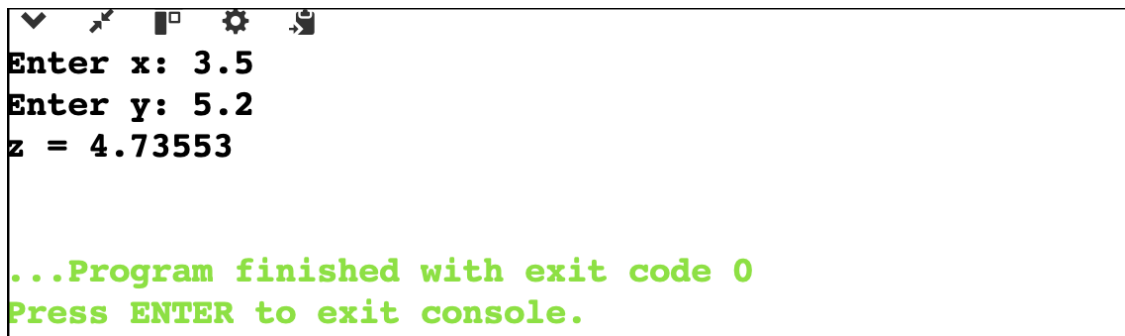
1. Умова: обчислити вираз, який визначається співвідношенням:

Під час розробки алгоритму врахувати, що змінні  $x$  і  $y$  користувач вводить з клавіатури, а результат обчислення змінної  $z$  повинен відобразитися на дисплеї.

2. Код програми на мові C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
double z, a1, a2, a3, x, y;
    cout<< "Enter x: ";
    cin>> x;
    cout<< "Enter y: ";
    cin>> y;
    a1 = pow(2, -x);
    a2 = sqrt(x + pow(fabs(y),(1./4.)));
    a3 = sqrt(exp(x - 1/sin(x)));
    z = a1 * a2 * a3;
    cout<< "z = " <<z<< "\n";
return 0;
}
```

3. Результат роботи програми представлено на рис. 4.



```
Enter x: 3.5
Enter y: 5.2
z = 4.73553

...Program finished with exit code 0
Press ENTER to exit console.
```

Рис. 4. Результат роботи програми

### Пояснення до програми:

У рядку коду: `double z, a1, a2, a3, x, y;`  
оголошені шість змінних `z, a1, a2, a3, x` і `y` дійсного типу

У рядку коду: `cout<< "Enter x: ";`  
за допомогою оператора `<<` і об'єкта `cout` відбувається звернення до потоку введення/виведення для виведення на дисплей напису `Enter x:` тобто запрошення користувача ввести значення змінної `x`.

У рядку коду: `cin >> x;`  
за допомогою оператора `>>` і об'єкта `cin` відбувається звернення до потоку введення/виведення для зчитування даних, що вводяться користувачем з клавіатури, і присвоєння їх змінній `x`.

У рядку коду: `cout<< "Enter y: ";`  
відбувається звернення до потоку введення-виведення і на дисплей виводиться напис `Enter y:` тобто запрошення користувачеві ввести значення змінної `y`.

У рядку коду `cin >> y;` відбувається звернення до потоку введення-виведення для зчитування даних, що вводяться користувачем з клавіатури та присвоєння їх змінній `y`.

У рядку коду: `a1 = pow(2, x);`  
відбувається обчислення допоміжної змінної `a1`.

У рядку коду: `a2 = sqrt(x + pow(abs(y), (1./4.)));`  
відбувається обчислення допоміжної змінної `a2`.

У рядку коду: `a3 = sqrt(exp(x - 1/sin(x)));`  
виконується обчислення допоміжної змінної `a3`

У рядку коду: `z = a1 * a2 * a3;`  
відбувається обчислення шуканого значення змінної `z`.

У рядку коду: `cout<< "z =" <<z<< "\n";`

виводиться обчислене значення змінної *z*. Символ `\n` означає перехід на наступний рядок.

### Контрольні запитання

1. З яких блоків складається програма на C++?
2. Для чого використовується інструкція `#include`?
3. Як працюють оператори `cin` та `cout`?
4. Які базові типи даних використовуються в C++?
5. Які арифметичні оператори використовуються в C++?
6. Що таке скорочена форма арифметичного оператора?
7. Які логічні оператори існують в C++?

### Лабораторна робота №2

#### РОЗРОБКА ПРОГРАМ З РОЗГАЛУЖЕННЯМИ. УМОВНІ ОПЕРАТОРИ

**Мета роботи:** навчитися розробляти алгоритми програм та програми, які містять умовні оператори.

#### Зміст лабораторної роботи

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти і відкоригувати програму рішення задачі.
3. Проаналізувати повну та скорочену конструкцію умовного оператора **If**.
4. Перевірити можливість використання оператора вибору **Switch** у програмі.
5. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
6. Скласти звіт.

#### Приклади розробки програм

*Приклад №1:*

1. Умова: дано дійсні числа *x, y, z*. Одержати  $\max(x, y, z)$ .

2. Графічна схема алгоритму показана на рис. 5.

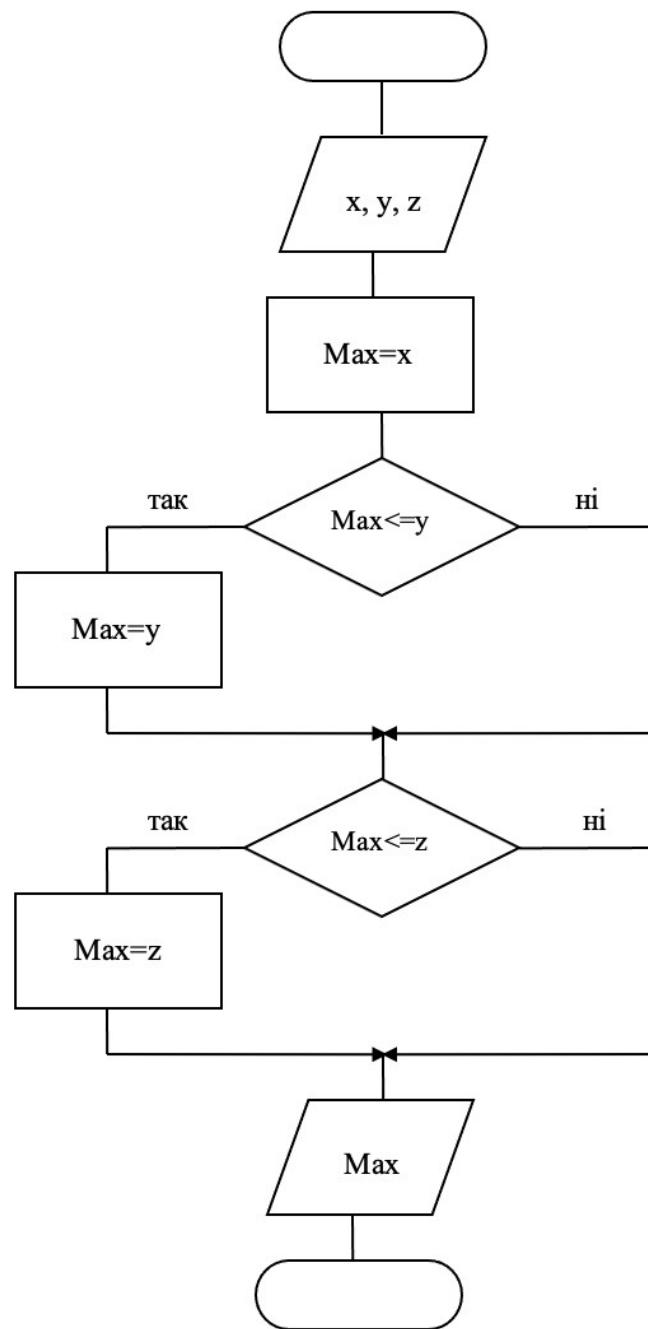


Рис. 5. Блок-схема

3. Код програми на мові C++ :

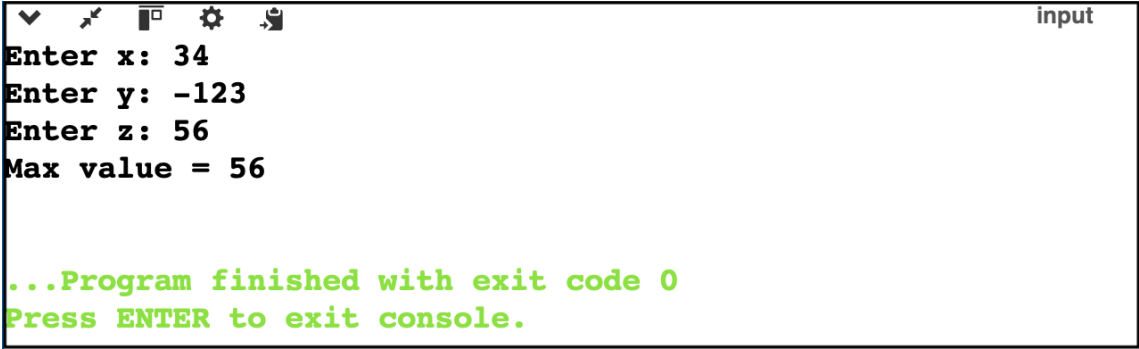
```
#include <iostream>
using namespace std;
int main()
{
```

```

double x, y, z, max;
  cout<< "Enter x: ";
  cin>> x;
  cout<< "Enter y: ";
  cin>> y;
  cout<< "Enter z: ";
  cin>> z;
  max=x;
  if (max<=y) max=y;
  if (max<=z) max=z;
  cout<< "Max value = " <<max<< "\n";
return 0;
}

```

4. Результат роботи програми представлено на рис. 6.



```

input
Enter x: 34
Enter y: -123
Enter z: 56
Max value = 56

...Program finished with exit code 0
Press ENTER to exit console.

```

Рис. 6. Результат виконання програми

**Пояснення до програми:** після введення числових значень змінних  $x$ ,  $y$  та  $z$ , за допомогою умовного оператора *If* знаходиться максимальне значення  $max$  і виводиться на екран оператором `cout<< "Max value = " <<max<< "\n"`.

*Приклад №2:*

1. Умова: дано дійсні числа  $x, y$ . Одержати результат залежно від обраної арифметичної дії –  $*$ ,  $/$ ,  $+$ ,  $-$ . Обробити ситуацію невірною коду арифметичної операції.
2. Код програми на мові C++ :

```
#include <iostream>
```

```

using namespace std;
int main()
{
double x, y, res;
char z;
    cout<< "Enter x: ";
    cin>> x;
cout<< "Enter y: ";
    cin>> y;
    cout<< "Operation: ";
    cin>> z;

    switch (z)
    {
        case '+': res=x+y;
            break;
        case '-': res=x-y;
            break;
        case '*': res=x*y;
            break;
        case '/': res=x/y;
            break;
        default : cout<< "Wrong code operation";
            return 0;
    }
    cout<< "Result = " <<res<< "\n";
return 0;
}

```

3. Результат виконання програми представлено на рис. 7 та рис. 8.



```

input
Enter x: 5
Enter y: 78
Operation: /
Result = 0.0641026

...Program finished with exit code 0
Press ENTER to exit console.

```

Рис. 7. Результат виконання програми



```
input
Enter x: 4
Enter y: 5
Operation: =
Wrong code operation

...Program finished with exit code 0
Press ENTER to exit console.
```

Рис. 8. Результат виконання програми

**Пояснення до програми:** після введення числових значень змінних  $x$ ,  $y$  за допомогою оператора вибору `switch() case()` застосовується до операндів введена користувачем арифметичну операцію. Мітка `default` буде виконуватися у випадку, коли введений символ арифметичної операції не співпадає з жодним із символів '+', '-', '\*', '/' та виведе на екран повідомлення про помилку. Результат обчислення виводиться на екран оператором `cout<< "Result = " <<res<< "\n";`.

### Контрольні запитання

1. В чому полягає особливість тернарного оператора? Наведіть приклади.
2. Чим відрізняється повна конструкція умовного оператора `if` від скороченої?
3. Як виглядає вкладеність умовного оператора `if`?
4. Який вигляд має оператор вибору `switch`?
5. В чому полягає принцип роботи оператора вибору `switch`?

## **Лабораторна робота №3**

### **СКЛАДАННЯ ПРОГРАМ З ВИКОРИСТАННЯМ ОПЕРАТОРІВ ЦИКЛУ**

**Мета роботи:** засвоїти методи використання операторів циклу.

#### **Зміст лабораторної роботи**

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти і відредагувати програму рішення задачі.
3. Проаналізувати роботу різних операторів циклу.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

#### **Приклади розробки програм**

*Приклад №1:*

1. Умова: обчислити .
2. Графічна схема алгоритму показана на рис. 9.

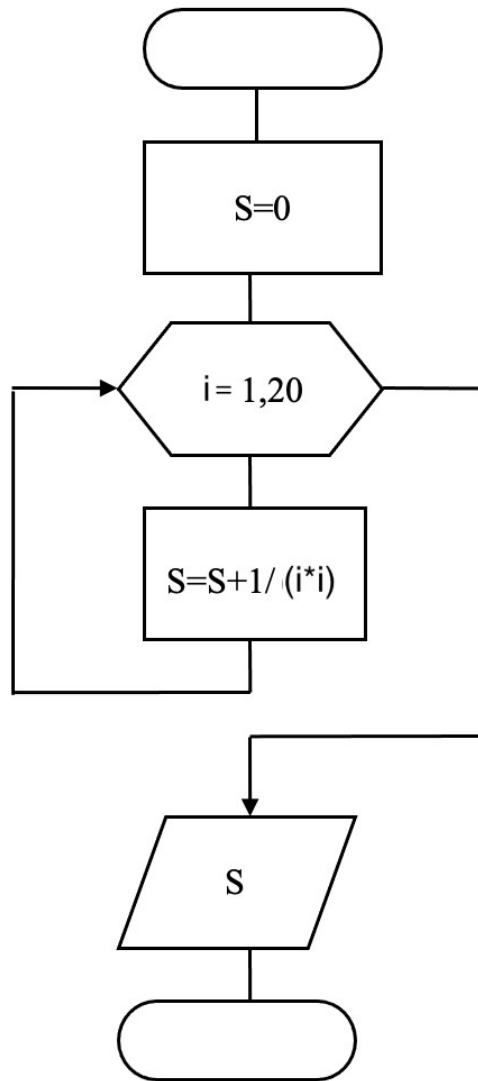


Рис. 9. Блок-схема

3. Код програми на мові C++:

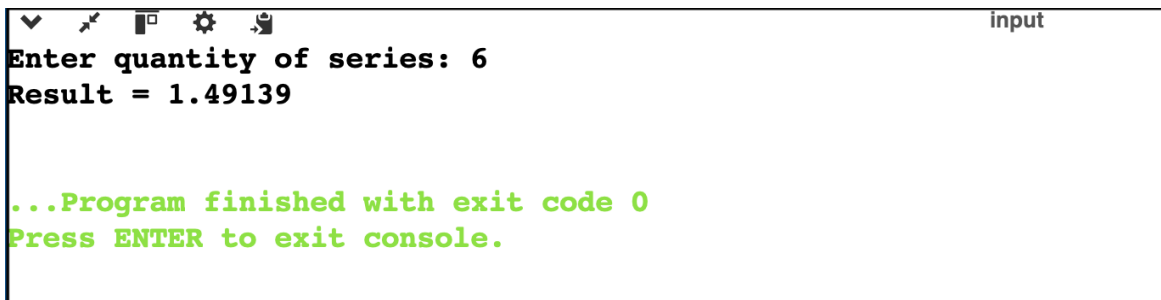
```

#include <iostream>
using namespace std;
int main() {
int n;
double sum=0;
    cout<< "Enter quantity of series: ";
    cin>> n;
    for(int i=1; i<=n; i++)
        sum=sum+1.0/(i*i);
    cout<< "Result = " <<sum<< "\n";
return 0;
}

```

}

4. Результат роботи програми представлено на рис. 10.

A screenshot of a console window titled 'input'. The window contains the following text: 'Enter quantity of series: 6', 'Result = 1.49139', and a green message: '...Program finished with exit code 0', 'Press ENTER to exit console.' The console window has a standard Windows-style title bar with icons for minimize, maximize, and close, and a system tray area with a volume icon and a network icon.

```
input
Enter quantity of series: 6
Result = 1.49139

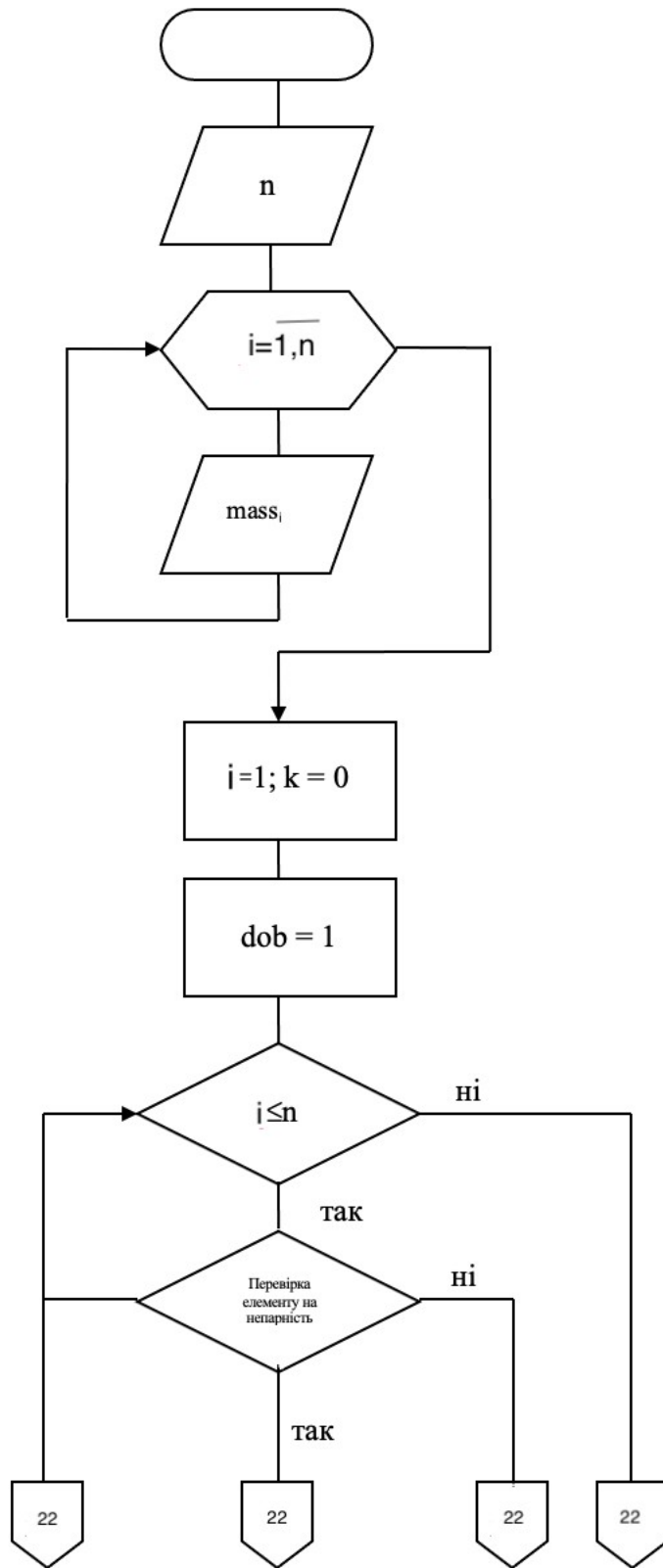
...Program finished with exit code 0
Press ENTER to exit console.
```

Рис. 10. Результат виконання програми

**Пояснення до програми:** під час обчислення суми початкове значення змінної  $s$ , де вона буде знаходитися, має дорівнювати 0. Далі за допомогою циклу *for* обчислюється сума, яка представлена відповідним виразом.

*Приклад №2:*

1. Умова: дано масив цілих чисел. Обчислити добуток та кількість непарних елементів масиву.
2. Графічна схема алгоритму наведена на рис. 11.



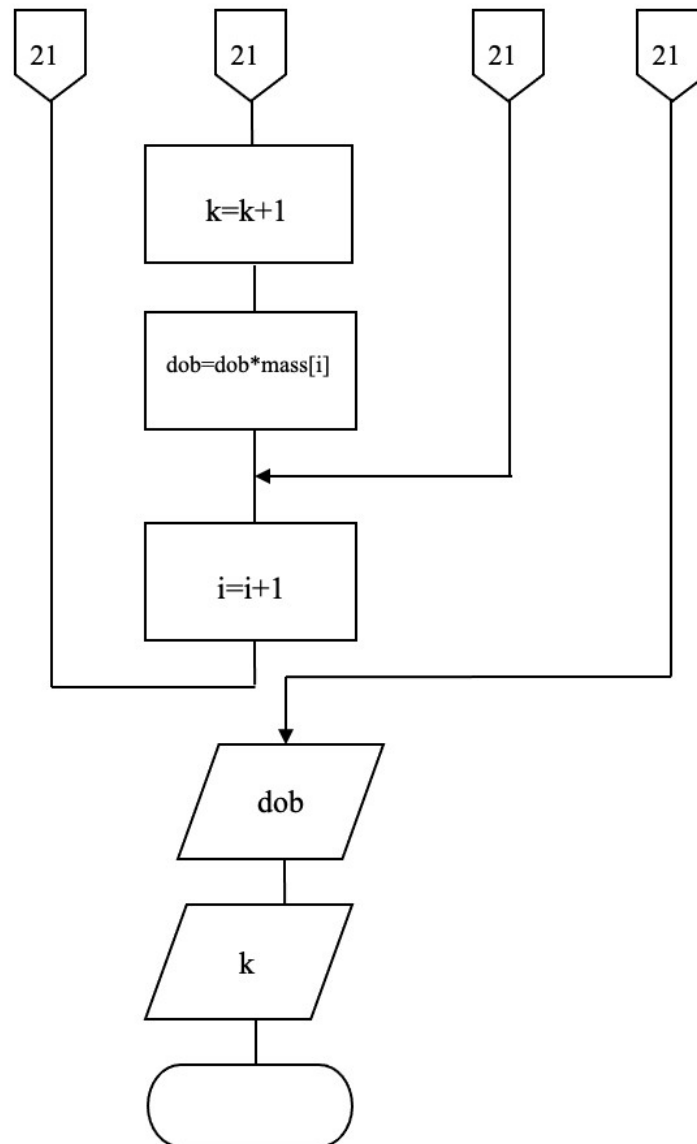


Рис. 11. Блок-схема

3. Код програми на мові C++ :

```

#include <iostream>
#include <ctime>

```

```

using namespace std;
int main()
{
int mass [100];
int k=0, n, dob=1;
    srand(time(0));
    cout<< "Enter quantity of elements: ";
    cin>> n;
    for (int i=0; i<n; i++)
        {
            mass[i]=rand()%15-3;
            cout<< "Enter element mass["<<i<<"] ="<<mass[i]<<endl;
        }
    int i=0;
    while (i<=n)
        {
            if (mass[i]%2!=0)
                {
                    k++;
                    dob=dob*mass[i];
                }
            i++;
        }
    cout<< "Quantity of the odd elements = "<<k<<endl;
    cout<< "Multiplication of the odd elements = "<<dob<<endl;
return 0;
}

```

4. Результат роботи програми представлено на рис. 12.

```
input
Enter quantity of elements: 7
Enter element mass[0]=11
Enter element mass[1]=8
Enter element mass[2]=0
Enter element mass[3]=-3
Enter element mass[4]=2
Enter element mass[5]=10
Enter element mass[6]=2
Quantity of the odd elements =2
Multiplication of the odd elements =-33

...Program finished with exit code 0
Press ENTER to exit console.
```

Рис. 12. Результат виконання програми

**Пояснення до програми:** на початку програми масив заповнюється випадковим чином за допомогою оператора *rand()*. Для оновлення бази випадкових чисел використовується функція *srand(time(0))*. За допомогою циклу *for* здійснюється введення масиву *mass[i]*. Перед циклом *while* змінній *k* необхідно присвоїти значення 0, а змінній *dob* присвоїти значення 1. Для того, щоб визначити непарний елемент масиву, використовується операція *%* (залишок від ділення). Функція *endl* здійснює перехід на новий рядок.

### Контрольні запитання

1. Який вигляд має конструкція циклу *for*?
2. Як може змінюватися значення параметра в циклі *for*?
3. Як працює оператор циклу *while*?
4. Чим відрізняється цикл *do...while* від циклу *while*?
5. Чому цикл *do...while* завжди виконується хоча б один раз?

### Лабораторна робота №4

#### РОЗРОБКА АЛГОРИТМІВ РІШЕННЯ МАТРИЧНИХ ЗАДАЧ

**Мета роботи:** вивчити методи розробки алгоритмів для вирішення задач з матрицями.

### Зміст лабораторної роботи

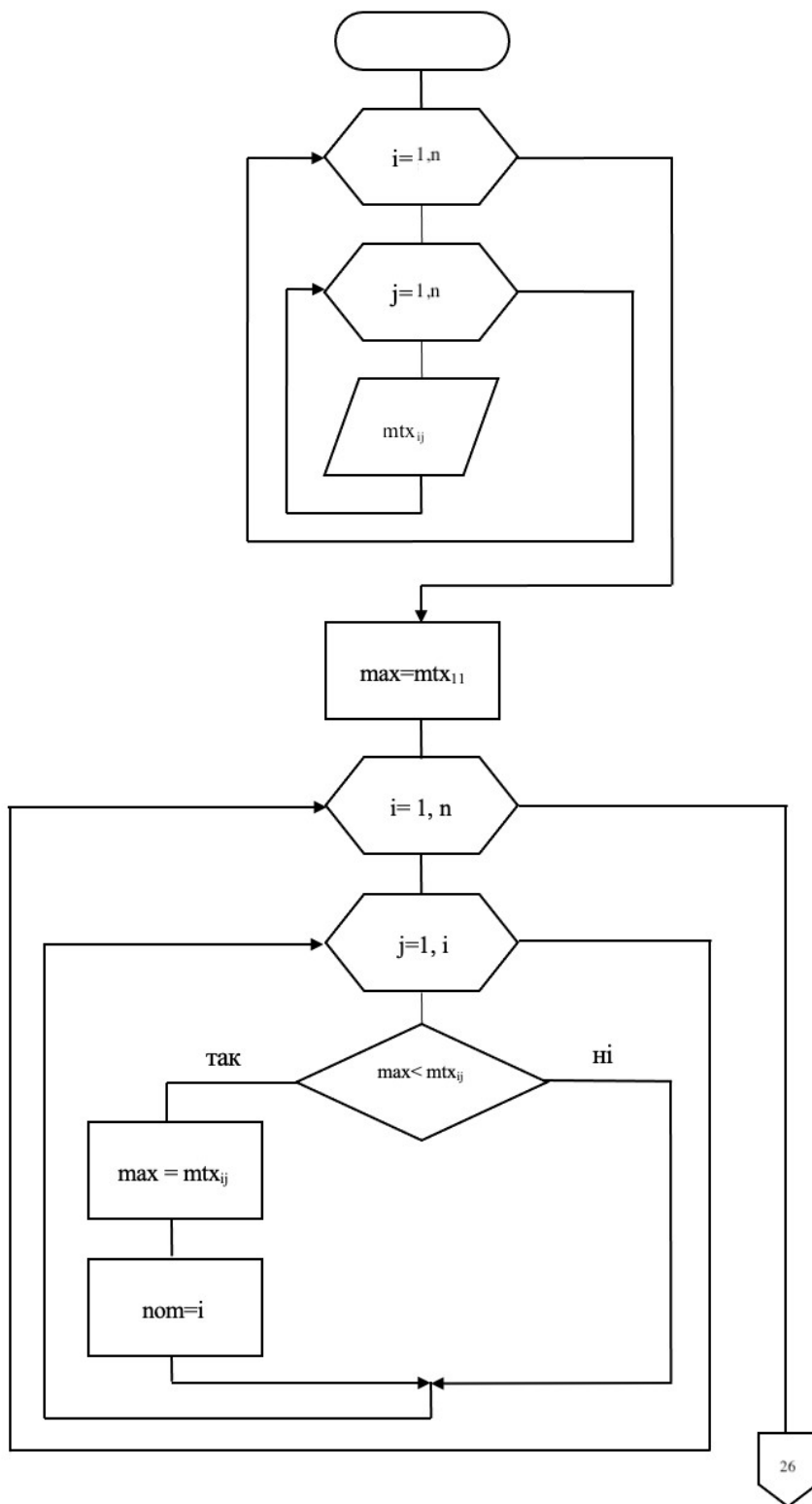
Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти і відкоригувати програму рішення задачі.
3. Проаналізувати методи, які використовувались під час розробки даної програми.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

### **Приклади розробки програм**

#### *Приклад №1:*

1. Умова: в квадратній матриці ліворуч від головної діагоналі знайти максимальний елемент і відповідний номер рядка.
2. Графічна схема алгоритму показана на рис. 13.



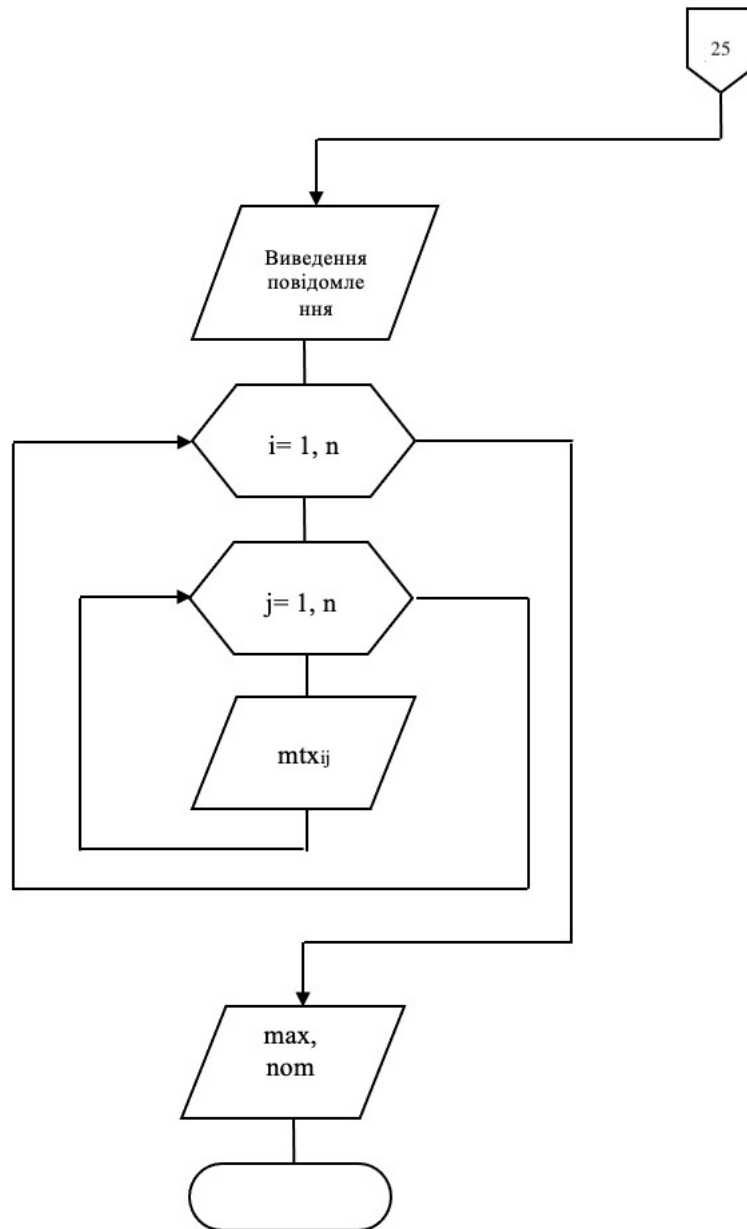


Рис. 13. Блок-схема

### 3. Код програми на мові C++ :

```

#include <iostream>
#include <ctime>
using namespace std;
int main()
{
const unsigned int max_rows=100;
const unsigned int max_columns=100;
int mtx [max_rows][max_columns];
int n, max, row;

```

```

cout<< "Enter size array: ";
    cin>>n;
srand(time(0));
cout<< "Matrix " <<n<<"x" <<n<<endl;
    for (int i=0; i<n;i++){
        for(int j=0; j<n;j++){
            mtx[i][j]=rand()%20-5;
            cout<< mtx[i][j]<<"\t";
        }
    }
cout<<endl;
}
cout<<endl;
max=mtx[0][0];
    for (int i=0; i<n;i++){
        for(int j=0; j<=i;j++){
            if (max<=mtx[i][j]){
                max=mtx[i][j];
                row=i;
            }
        }
    }
cout<<endl;
cout<< "Max element in input array = " <<max<<endl;
cout<< "The row number " <<row+1<<" has max element of input
array." <<endl;
return 0;
}

```

4. Результат роботи програми представлено на рис. 14.

```
Enter size array: 5
Matrix 5x5
-4      9      2      -2      7
13      6      8      -5      1
14      2     -1     13      4
4       -2      0      -3     12
9        1      6     12      6

Max element in input array = 14
The row number 3 has max element of input array.

...Program finished with exit code 0
Press ENTER to exit console.
```

Рис. 14. Результат виконання програми

**Пояснення до програми:** на початку програми здійснюється введення елементів матриці за допомогою двох циклів. Важливе місце в цій програмі займає організація внутрішнього циклу `for (int j=0; j<=i; j++){...}`. Він забезпечує перегляд всіх елементів, які розміщені на головній діагоналі та ліворуч від неї і пошук максимального елемента серед них.

Наприкінці програми розташовані оператори, які забезпечують виведення матриці, значення максимального елемента та номера рядка, в якому він знаходиться.

### Контрольні запитання

1. Як описати масив у програмі?
2. Змінні якого типу можуть бути значеннями індексів масиву?
3. Як здійснюється введення та виведення одновимірному масиву?  
Навести приклади.
4. Які існують методи сортування масивів?
5. Як можна описати і заповнити двовимірний масив?

### Лабораторна робота №5

#### РОЗРОБКА ПРОГРАМ З ВИКОРИСТАННЯМ ФУНКЦІЙ

**Мета роботи:** вивчити правила та методи організації функцій у C++.

#### Зміст лабораторної роботи

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.

2. Скласти програму, яка містить функцію і відкоригувати її.
3. Проаналізувати види функцій, які використовувались під час розробки даної програми.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

### Приклади розробки програм

*Приклад №1:*

1. Умова: визначити максимальний елемент в різних масивах.
2. Графічна схема алгоритму наведена на рис. 15.

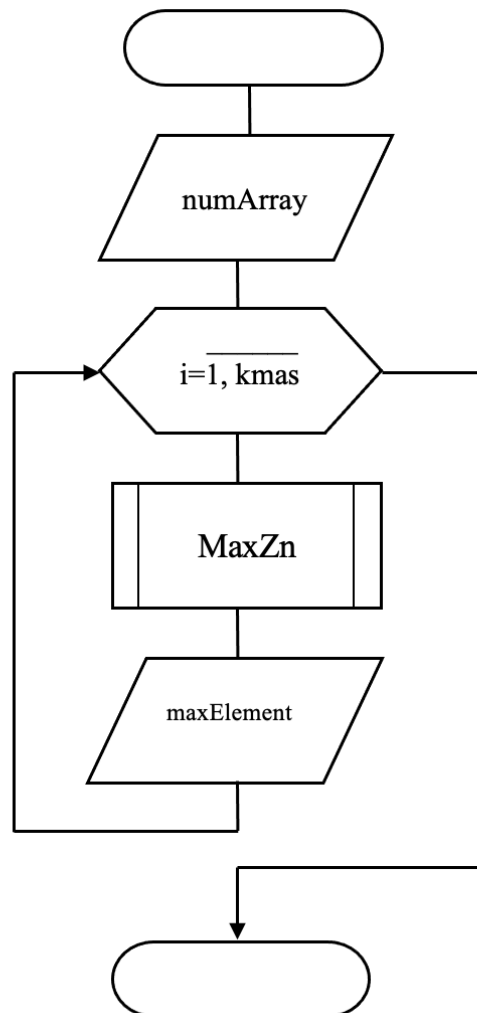


Рис. 15. Блок-схема головного коду програми

3. Код програми на мові C++:

```
#include <iostream>
```

```

using namespace std;
void MaxZn (int numArray){
    int numberElements;
    float temporyArray[100];
    float maxElement;
    cout << "Enter the number of elements in the array ";
    cin >> numberElements;
    for (int i = 0; i<numberElements; i++)
        {
            cout <<i+1<< "Element of the array number " << numArray+1<<" :
";
            cin >> temporyArray[i];
        }
    maxElement = temporyArray[0];

    for (int i = 1; i<numberElements; i++)
        {
            if (maxElement<temporyArray[i]) { maxElement =
temporyArray[i]; }
        }
    cout << "Maximum element of the array number " << numArray+1 << "
is " << maxElement <<endl<<endl;
}
int main()
{
    int numberArray;
    cout << "Enter quantity of arrays ";
    cin >> numberArray;
    for (int i = 0; i<numberArray; i++){ MaxZn (i); }
    return 0;
}

```

4. Результат виконання програми представлено на рис. 16.

```

Enter quantity of arrays:3
Enter the number of elements in the array:5
1 Element of the array number 1:4
2 Element of the array number 1:-5
3 Element of the array number 1:6
4 Element of the array number 1:7
5 Element of the array number 1:-10
Maximum element of the array number 1 is 7

Enter the number of elements in the array:6
1 Element of the array number 2:57
2 Element of the array number 2:-34
3 Element of the array number 2:-78
4 Element of the array number 2:5
5 Element of the array number 2:9
6 Element of the array number 2:12
Maximum element of the array number 2 is 57

Enter the number of elements in the array:4
1 Element of the array number 3:34
2 Element of the array number 3:-90
3 Element of the array number 3:23
4 Element of the array number 3:11
Maximum element of the array number 3 is 34

```

Рис. 16. Результат виконання програми

**Пояснення до програми:** у функції `MaxZn` створюється масив `temporyArray`, який може містити дані типу `float`. Кількість елементів в цьому масиві конкретизується перед його введенням і аналізом. Результат роботи функції друкується у функції `MaxZn`.

*Приклад №2:*

1. Умова: обчислення факторіалу  $n!$ .
2. Код програми на мові C++:

```

#include <iostream>
using namespace std;
int Fact (int numb)
{
    long int rezult = 1;
    for (int i = 1; i <= numb; i++) { rezult *= i; }
    return rezult;
}

```

```

int main()
{
    int digit;
    cout << "Enter number: ";
    cin >> digit;
    cout << digit << "! = " << Fact(digit) << endl;
    return 0;
}

```

3. Результат виконання програми представлено на рис. 17.

```

Enter number:15
15! = 2004310016

...Program finished with exit code 0
Press ENTER to exit console.

```

Рис. 17. Результат виконання програми

**Пояснення до програми:** функція *Fact* обчислює факторіал числа *n*. Проміжні результати передаються в стек і там зберігаються до останнього кроку.

### Контрольні запитання

1. Дайте визначення функції.
2. Які види функцій існують в мові програмування C++?
3. Яка існує різниця між передачею аргументів за значенням від передачі аргументів через посилання?
4. Як відбувається передача масиву аргументом функції?
5. Що таке прототип функції?
6. Що таке рекурсія? Навести приклади.
7. Що означає поняття «перевантаження функцій»?
8. Дайте визначення поняттю «аргументи за замовчуванням».

## Лабораторна робота №6

### РОЗРОБКА ПРОГРАМ З ВИКОРИСТАННЯМ СИМВОЛЬНИХ ЗМІННИХ

**Мета роботи:** вивчити особливості виконання функцій для обробки символьних даних в мові C++.

#### Зміст лабораторної роботи

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти програму і відредагувати її.
3. Проаналізувати види підпрограм, які використовувались для обробки текстових даних під час розробки програми.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

#### Приклади розробки програм

1. Умова: дано текст. Відомо, що в тексті слова розділені символом пробіл, а текст завершується крапкою. Визначити кількість слів у тексті.
2. Код програми на мові C++:

```
#include <string>
#include <iostream>
using namespace std;
int main(){
    string s;
    int quantity=0;
    cout << "Enter string:";
    getline(cin, s);
    if (s[s.size()-1] != '.')
    {
        cout << "There is no point at the end of the sentence"<<endl;
        return 0;
    }
}
```

```

for (int i = 0; i < s.size(); i++){
    if (s[i] == ' ') { quantity++; }
    cout << "The number of words is " << quantity+1<<endl;
return 0;
}

```

3. Результат виконання програми представлено на рис. 18.

```

Enter string:The sky is blue.
The number of words is 4

...Program finished with exit code 0
Press ENTER to exit console.

```

Рис. 18. Результат виконання програми

**Пояснення до програми:** введення тексту здійснюється командою `getline(cin, s)`, що забезпечує зчитування слів, які розділені пробілами. На наступному кроці визначається загальна довжина тексту. Для цього використана функція `s.size()`.

У програмі передбачена перевірка наявності в кінці речення крапки. Якщо крапка в кінці речення не була введена, програма припиняє свою роботу і на екрані з'являється відповідне повідомлення.

Цикл *for* забезпечує перебір всіх символів рядка та визначення символу «пробіл», який розділяє окремі слова. В циклі визначається кількість слів за допомогою оператора *If*.

### Контрольні запитання

1. Як описуються символні змінні в мові C++?
2. За допомогою якої функції визначається довжина рядка?
3. Як здійснюється зчеплення рядків?
4. Які особливості виконання функцій `atoi()`, `atof()`?
5. Як порівняти два рядки? Навести приклад.
6. Як працюють функції `tolower()` і `toupper()`?
7. Які особливості виконання функції `strtok()`?

### Лабораторна робота №7

# БІТОВІ ОПЕРАЦІЇ ТА ОСОБЛИВОСТІ ЇХ РЕАЛІЗАЦІЇ

**Мета роботи:** вивчити методи обробки інформації, що представлена в двійковій формі.

## Зміст лабораторної роботи

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти і відкоригувати програму рішення задачі.
3. Проаналізувати методи, які використовувались під час розробки даної програми.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

## Приклади розробки програм

*Приклад №1:*

1. Умова: є два двійкових слова `Low_Byte`, `High_Byte`, довжина яких складає 8 біт. В двійковому слові `High_Byte` інформаційними є  $n$  розрядів (старших або молодших). В двійковому слові `Low_Byte` усі розряди є інформаційними. Необхідно із слів `Low_Byte` і `High_Byte` утворити одне слово, довжиною  $(n+8)$  біт, в якому старші біти відповідають значенням  $n$  інформаційних бітів в `High_Byte`, а молодші 8 біт співпадають з `Low_Byte`. У процесі перевірки отриманого програмного коду враховувати, що значення отриманого результату не повинно перевищувати  $2^{n+8}$ .
2. Код програми на мові C++ :

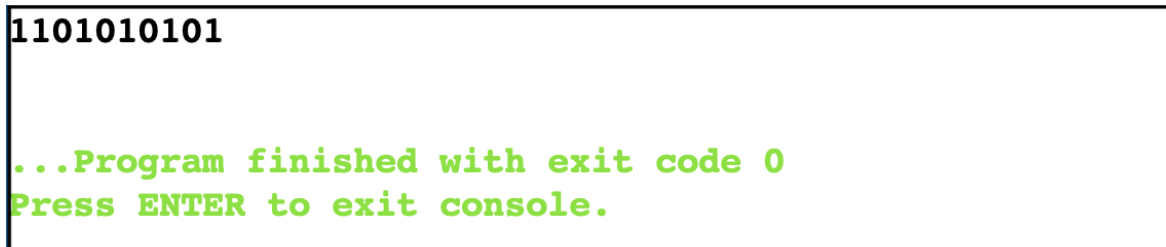
```
#include <iostream>
#include <cstdlib>
#include <bitset>
using namespace std;
int main(){
    int Low_Byte = 85;
    int High_Byte = 3;
    int Result=0;
```

```

Result = High_Byte << 8 | Low_Byte;
cout << bitset<10>(Result) << endl;
    return 0;
}

```

3. Результат виконання представлено на рис. 19.



```

1101010101
...Program finished with exit code 0
Press ENTER to exit console.

```

Рис. 19. Результат виконання програми

**Пояснення до програми.** Отримати двійкове слово з  $(n+8)$  розрядами вдається, якщо застосувати операцію зсуву вправо або вліво (залежить від того, які розряди в `High_Byte` є інформаційними), та операцію порозрядного АБО (позначається `|`).

Операція зсуву забезпечує зсув усіх розрядів змінної на задану кількість позицій. Кінцевий результат отримаємо в результаті виконання операції порозрядного АБО між отриманим на попередньому кроці  $n+8$  розрядним проміжним результатом і байтом `Low_Byte`, що визначає 8 молодших розрядів.

Відзначимо, що введення чисел в двійковому зображенні в C++ не передбачено. Тому значення змінних `Low_Byte`, `High_Byte` та `Result` задано в десятковій формі (проте є можливість записувати числа в системі числення з основою 16 або 8). Для виводу числа в двійковій формі використано спеціальний клас-контейнер для зберігання бітів – `bitset`.

*Приклад №2:*

1. Умова: розробити програму, що буде утворювати 8-розрядне слово із бітів, значення яких задано.
2. Код програми на мові C++ :

```

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

int main()
{

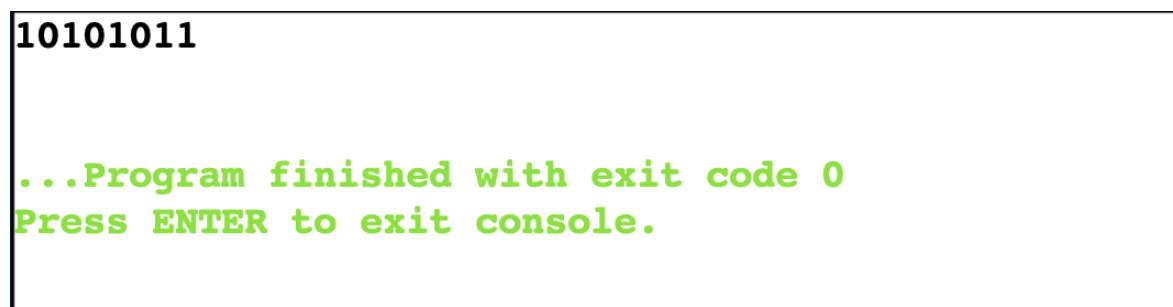
```

```

int bit_0 = 1;
int bit_1 = 1;
int bit_2 = 0;
int bit_3 = 1;
int bit_4 = 0;
int bit_5 = 1;
int bit_6 = 0;
int bit_7 = 1;
int Result = 0;
Result = Result | bit_7 << 7 | bit_6 << 6 | bit_5 << 5 | bit_4 << 4 | bit_3 <<
3 | bit_2 << 2 | bit_1 << 1 | bit_0;
cout << bitset<8>(Result) << endl;
return 0;
}

```

3. Результат виконання представлено на рис. 20.



```

10101011
...Program finished with exit code 0
Press ENTER to exit console.

```

Рис. 20. Результат виконання програми

**Пояснення до програми.** Розв’язок поставленої задачі ґрунтується на розглянутому раніше підході, в основу якого покладено використання операції зсуву та операції побітового АБО. Щоб утворити 8-розрядне слово, в якому два старших розряди визначаються значеннями змінних bit\_7 та bit\_6, потрібно bit\_7 зсунути вліво на 7 позицій, а bit\_6 – на 6 позицій. Аналогічно зсув вліво на відповідну кількість позицій слід виконати для змінних bit\_5 – bit\_1. Кінцевий результат отримуємо після виконання над отриманими проміжними результатами операції порозрядного АБО.

### Контрольні запитання

1. Які бітові операції існують в C++?
2. Яка бітова операція реалізує операцію ділення числа на 2?

3. Яка бітова операція реалізує операцію множення числа на 2?
4. Як за допомогою маски виділити задані біти в байті?
5. Як в C++ задаються двійкові числа?
6. Як вивести у вікно консолі двійкове число?

## Лабораторна робота №8 СТРУКТУРИ

**Мета роботи:** розглянути особливості структур і навчитися розробляти програми, використовуючи цей тип даних.

### Зміст лабораторної роботи

Перед виконанням лабораторної роботи студентам необхідно вивчити відповідні розділи конспекту лекцій та рекомендованої літератури.

1. Проаналізувати алгоритм рішення задачі відповідно до варіанта завдання.
2. Скласти програму і відредагувати її.
3. Проаналізувати оператори під час роботи з записами, які використовувались у процесі розробки програми.
4. Вивести на екран результати роботи програми, проаналізувати їх та зробити висновки.
5. Скласти звіт.

### Приклад розробки програми

*Приклад:*

1. Умова. Дано: група студентів з п'яти осіб. Анкетні дані: прізвище, рік народження та оцінки за останню сесію. Визначити кількість відмінників.
2. Код програми на мові C++:

```
#include <iostream>
using namespace std;
int main(){
    struct student {
        char SN[30];
        char db[8];
        int marks[4];
    } stud[100];
    int c_st;
    cout << " Enter number of students: ";
```

```

cin >> c_st;
for (int i = 0; i < c_st; ++i){
    cout << "Enter surname:";
    cin >> stud[i].SN;
    cout << " Enter date of birth: ";
    cin >> stud[i].db;
    cout << "Four marks of student:";
    for (int j = 0; j < 4; ++j){
        stud[i].marks[j]=rand()%3+3;
        cout<< stud[i].marks[j]<<" ";
    }
    cout << endl;
}
int c_st_5 = 0, k;

for (int i = 0; i < c_st; ++i){
    k = 0;
    for (int j = 0; j < 4; ++j)
        if(stud[i].marks[j] == 5)k++;
    if (k == 4) {cout << "Good students";
        cout << stud[i].SN << " " << stud[i].db;
        c_st_5++;
    }
    cout << endl;
}
if (c_st_5 == 0) cout << "Such students don't exist! "; else
cout << "Number of good students: " << c_st_5 << endl;
return 0;}

```

3. Результат виконання представлено на рис. 21.

```

Enter number of students:4
Enter surname:Petrenko
Enter date of birth: 23/09/2005
Four marks of student:4 4 3 4
Enter surname:Sidorenko
Enter date of birth: 24/06/2005
Four marks of student:5 4 4 3
Enter surname:Puliy
Enter date of birth: 14/07/2006
Four marks of student:3 4 5 4
Enter surname:Simonenko
Enter date of birth: 15/10/2006
Four marks of student:5 4 5 4
Such students don't exist!

```

Рис. 21. Результат виконання програми

**Пояснення до програми:** створюється структура `student`. Користувачу пропонується ввести кількість студентів з клавіатури. За допомогою циклу `for()` вводяться відповідні поля структури. Далі визначається кількість «5» серед оцінок кожного студента. Якщо кількість «5» дорівнює кількості предметів, прізвище відмінника виводиться на екран і обчислюється їх кількість.

### Контрольні запитання

1. Що таке «структура»?
2. Як описується тип даних «структура»?
3. Що називається полем «структури»?
4. Як отримати доступ до поля «структури»?

### СПИСОК ЛІТЕРАТУРИ

1. *Програмування на C++ у прикладах та задачах* : посібник / А.Н. Васильєв. – Київ : Видавництво «Ліра-К», 2019. – 382 с.
2. *Beginning C++23 : From Beginner to Pro 7th ed. Edition.* / Ivor Horton, Peter Van Weert – Apress, 2023. – 948 p.
3. *Tour of C++* / Bjarne Stroustrup. – [Addison-Wesley Professional](https://www.addison-wesley.com/), 2018. – 256 p.
4. *Задачи по программированию* : пособие / С.А. Абрамов, Г.Г. Гнездилова, Е.Н. Капустина, М.И. Селюк. – М.: Наука. Гл. ред. Физ.-мат. лит., 1999. – 224 с.

Навчально-методичне видання

## КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ТА ПРОГРАМУВАННЯ

Методичні вказівки  
до виконання лабораторних робіт

для здобувачів першого (бакалаврського) рівня вищої освіти  
за спеціальностями 174 «Автоматизація, комп'ютерно-інтегровані  
технології та робототехніка» та 141 «Електроенергетика, електротехніка  
та електромеханіка»

Укладачі: **Бондарчук** Ольга Вячеславівна,  
**Луценко** Вадим Юрійович

Випусковий редактор *Л. С. Тавлуй*  
Комп'ютерне верстання *К. А. Мавроді*

Підписано до друку 11.09.2025. Формат 60 x 84<sub>1/16</sub>  
Ум. друк. арк. 2,32. Обл.-вид. арк. 2,5.  
Електронний документ. Вид. № 48/Ш-25

Видавець і виготовлювач:  
Київський національний університет будівництва і архітектури  
Проспект Повітряних Сил, 31, Київ, Україна, 03037

Свідоцтво про внесення до Державного реєстру суб'єктів  
видавничої справи ДК № 808 від 13.02.2002  
Свідоцтво про внесення до Державного реєстру суб'єктів  
видавничої справи ДК № 808 від 13.02.2002 р.