

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Л.В. Соловей, Н.М. Мірошніченко, Т.Г. Бабак

ПРОГРАМУВАННЯ МОВОЮ C# Visual Studio 2019

Лабораторний практикум

для студентів хімічних спеціальностей усіх форм навчання

У трьох частинах

Частина 1

ЗАТВЕРДЖЕНО
редакційно-видавничою
радою університету,
протокол № 1 від 28.01.2022 р.

Харків
НТУ «ХПІ»
2022

УДК 044.43 (076.5)

С 60

Р е ц е н з е н т и:

В. М. Колодяжний, д-р фіз.-мат. наук, професор ХНАДУ;

Є. Д. Пономаренко, доцент НТУ «ХПІ»

Соловей Л.В.

С 60 Програмування мовою С# Visual Studio 2019. Лабораторний практикум: у 3-х ч. Ч. 1. / Л. В. Соловей, Н. М. Мірошніченко, Т. Г. Бабак. – Харків : НТУ «ХПІ», 2022. – 120 с.

Лабораторний практикум присвячений вивченню мови програмування С# Visual Studio.NET 2019. Наведено велику кількість прикладів написання програм різної складності. До прикладів надаються пояснення. Усі програми забезпечені результатами виконання. До кожної теми подані практичні завдання для виконання лабораторних робіт. Матеріал розміщено за принципом поступовості, починаючи з основ до більш складних можливостей мови програмування С #.

Призначено для студентів хімічних спеціальностей, у тому числі для іноземних студентів.

Ил. 82 Табл. 7 Бібліогр. 3

УДК 044.43 (076.5)

© Соловей Л.В., Мірошніченко Н.М., Бабак Т.Г., 2022

© НТУ «ХПІ», 2022

ЗМІСТ

РІШЕННЯ ЗАДАЧ У ВИГЛЯДІ КОНСОЛЬНОГО ДОДАТКА

Вступ	4
Тема 1. СКЛАДАННЯ ПРОГРАМ ЛІНІЙНОЇ СТРУКТУРИ	5
1.1 Приклади розв'язання задач	5
1.2 Завдання для виконання лабораторних робіт (частина 1)	23
1.3 Завдання для виконання лабораторних робіт (частина 2)	24
Тема 2. СКЛАДАННЯ ПРОГРАМ РОЗГАЛУЖЕНОЇ СТРУКТУРИ	28
2.1 Приклади розв'язання задач	28
2.2 Завдання для виконання лабораторних робіт (частина 1)	44
2.3 Завдання для виконання лабораторних робіт (частина 2)	45
Тема 3. ЦИКЛИ	47
3.1 Приклади розв'язання задач	47
3.2 Завдання для виконання лабораторних робіт (частина 1)	59
3.3 Завдання для виконання лабораторних робіт (частина 2)	60
Тема 4. Одновимірні масиви	63
4.1 Приклади розв'язання задач	63
4.2 Завдання для виконання лабораторних робіт (частина 1)	89
4.3 Завдання для виконання лабораторних робіт (частина 2)	91
Тема 5. Двовимірні масиви	93
5.1 Приклади розв'язання задач	93
5.2 Завдання для виконання лабораторних робіт	116
Список літератури	119

Вступ

Існує чимало мов програмування, але лише деякі з них дійсно вдалі. Гарна мова програмування повинна бути одночасно ефективною і гнучкою, а її синтаксис – стислим, але ясным. Він повинен підтримувати найсучасніші можливості програмування. Саме такою мовою і є C#.

Мова C# була створена корпорацією Microsoft для підтримки середовища .NET Framework і спирається на багату спадщину в сфері програмування. Її головним розробником був Андерс Хейльсберг (Anders Hejlsberg) – відомий фахівець з програмування.

C# походить безпосередньо від двох найбільш вдалих у сфері програмування мов: C і C++. Від мови C він успадкував синтаксис, багато ключових слів і оператори, а від C++ – вдосконалену об'єктну модель. Крім того, C# тісно пов'язаний з Java – іншою не менш вдалою мовою.

Завдяки своїй здатності швидко пристосовуватися до постійно мінливих потреб у сфері програмування, C#, як і раніше, залишається живою і новаторською мовою. А, отже, вона є однією з найбільш ефективних і багатих на свої можливості мов у сучасному програмуванні.

У мові C# вдало поєднуються випробовані засоби програмування з останніми нововведеннями, і надається можливість для ефективного написання програм, призначених для обчислювального середовища сучасних підприємств. Це, без сумніву, одна з найважливіших мов програмування XXI століття.

Лабораторний практикум складається з трьох частин. У першій частині і в другій приклади являють собою консольні додатки, а в третій - Windows-додатки.

У лабораторному практикумі по кожній темі наведені приклади розв'язання задач і варіанти для самостійної роботи.

У першій частині розглянуті теми: «Складання програм лінійної структури», «Складання програм розгалуженої структури», «Цикли», «Одновимірні масиви», «Двовимірні масиви».

У другій частині наведено вирішення завдань на теми: «Рядки», «Методи», «Класи», «Структури, перерахування», «Файли».

У третій частині розглянуті приклади створення Windows-додатків.

Тема 1. СКЛАДАННЯ ПРОГРАМ ЛІНІЙНОЇ СТРУКТУРИ

Мета роботи: навчитись розробляти консольні додатки, вивчити оператори оголошення і ініціалізації змінних і констант; оператори присвоювання, коментарі; методи введення/виведення: `Console.Write()`, `Console.WriteLine()`, `Console.Read()`, `Console.ReadLine()`

1.1 Приклади розв'язання задач

Задача 1.1. Розробка консольних додатків.

Створимо консольний додаток (консоль зазвичай мають на увазі як екран комп'ютера і клавіатуру), який виводить результат обчислень у консольне вікно. Іноді для наукових розрахунків потрібно організувати найпростіше уведення даних, виконати складну математичну обробку уведених даних і оперативно вивести на екран результат обчислень. У такому випадку програмується так званий консольний додаток.

Створення консольного додатка :

1. Завантажити систему *Microsoft Visual Studio 2019*.

Після завантаження *Visual Studio* відображається вікно, основний простір якого займає початкова сторінка *Microsoft Visual Studio*, що містить команди створення і відкриття проектів, посилання на останні проекти, довідкові матеріали, розширення *Visual Studio*, ресурси співтовариства.

Початковий інтерфейс користувача наведено на рис. 1.1.

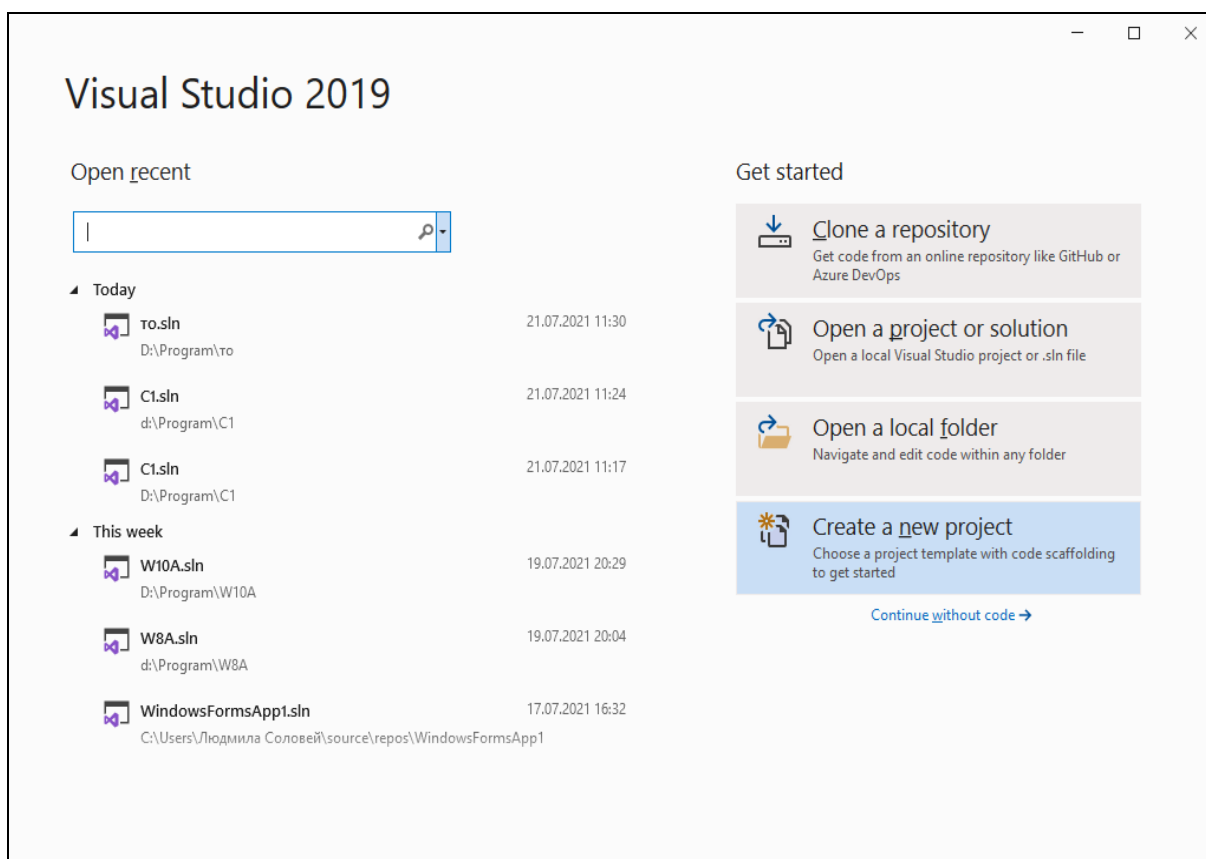


Рисунок 1.1 – Початковий інтерфейс користувача Visual C# 2019

2. Створити новий проект.

Викликати команду *Create a new project*.

У вікні *Create a new project* (рис. 1.2) зверху є кілька випадаючих списків:

All languages (Усі мови). В ньому знаходиться список інсталюваних шаблонів (Installed Templates). Серед них – шаблони мов програмування, вбудованих у Visual Studio, в т.ч. Visual Basic, Visual C#, Visual C++ та ін.

Вибираємо мову C#.

All platforms (Всі платформи) – Windows

All Project type(Всі типи проектів) – Console

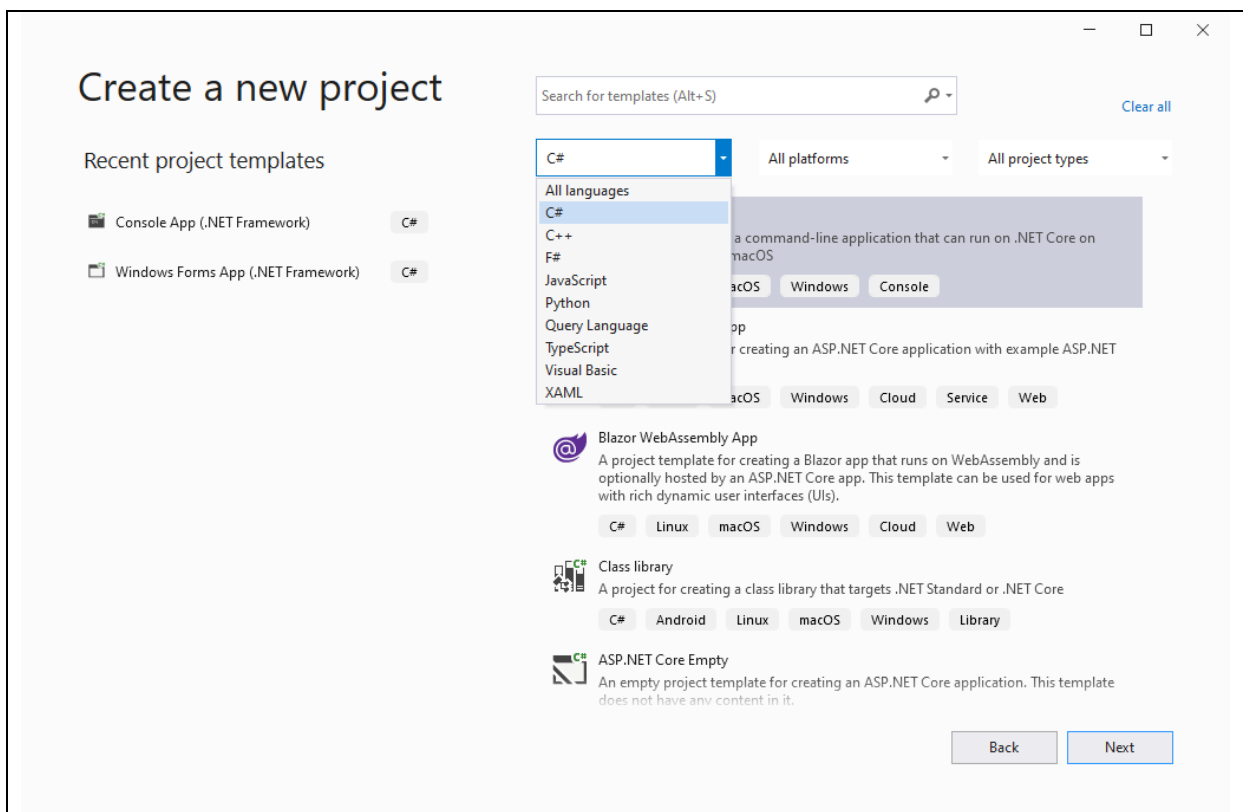


Рисунок 1.2 – Вікно *Create a new project*

Нижче у вікні *Create a new project* виберіть проект *Console App (.NET Framework)* (рис. 1.3).

У новому вікні *Configure your new project* (рис. 1.4) задаємо:

- у полі *Project name* введіть ім'я проекту – C1 (за замовчуванням система задає ім'я *ConsoleApp1*

- у полі *Location* (розташування) вкажіть шлях за замовчуванням, куди буде записаний проект. Внесіть у поле шлях *d:\Program*.

- у полі *Framework* (платформа) буде вказана платформа за замовчуванням: версія .NET Framework 4.7.2.

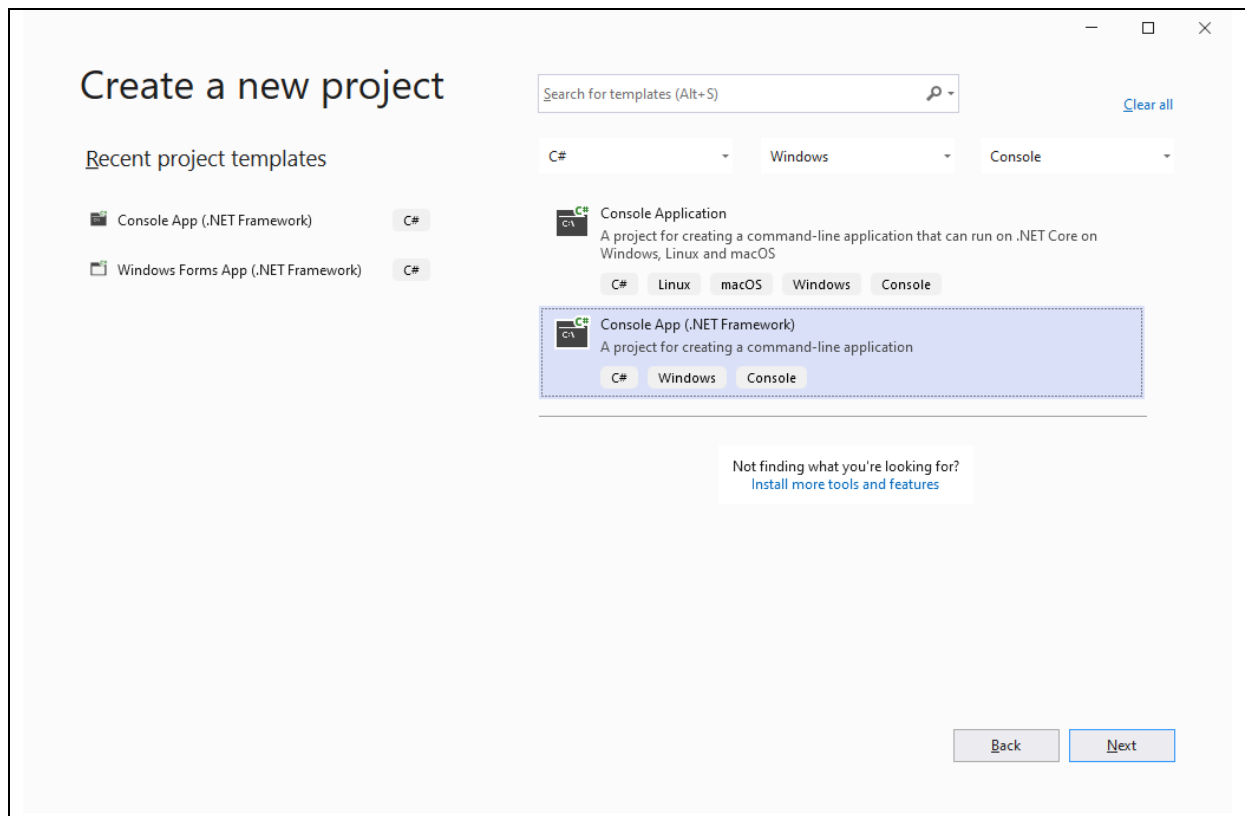


Рисунок 1.3– Вікно *Create a new project*

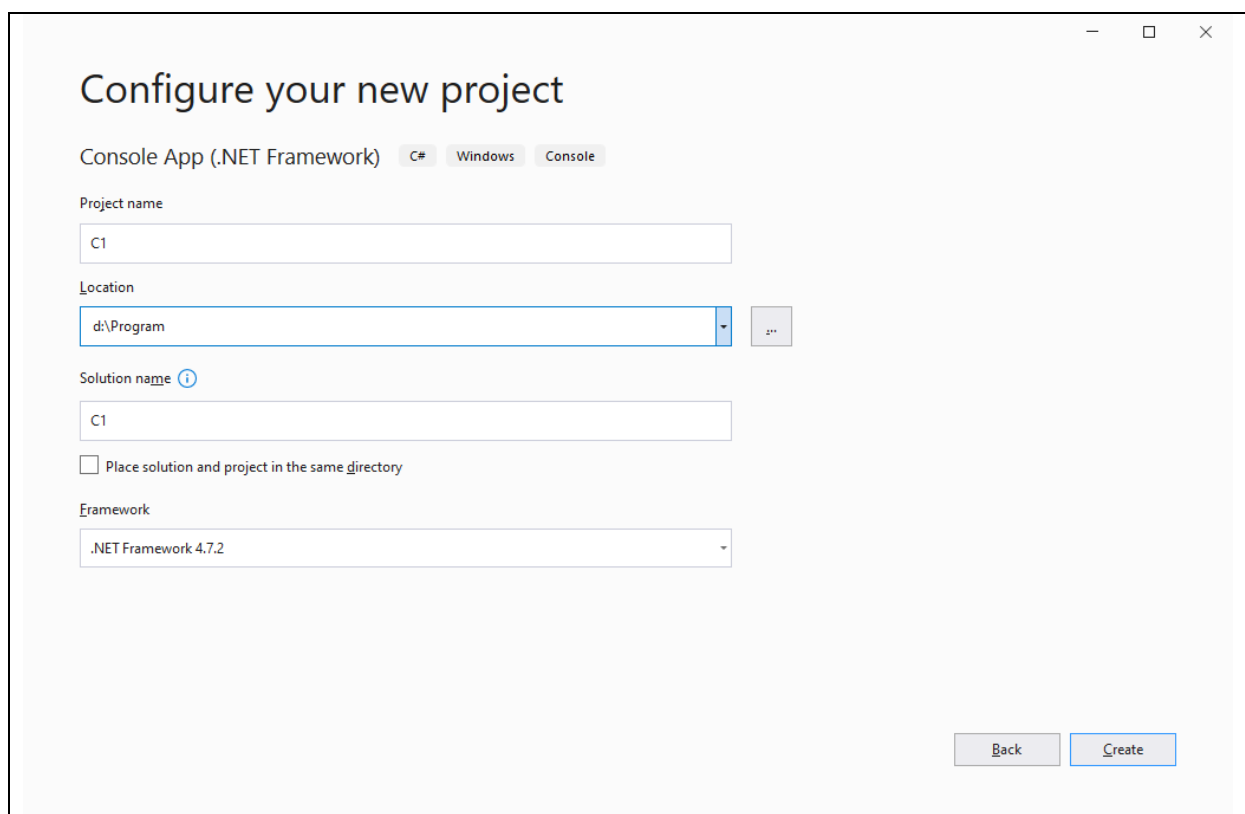


Рис. 1.4 – Вікно *Configure your project*

У теці проектів при створенні проекту створюється відповідна структура тек, що містить компоненти проекту.

Після натиснення клавіші *Create* потрапляємо відразу на вкладку програмного коду. У вікні редактора з'явиться згенерований автоматично каркас програми (рис. 1.5).

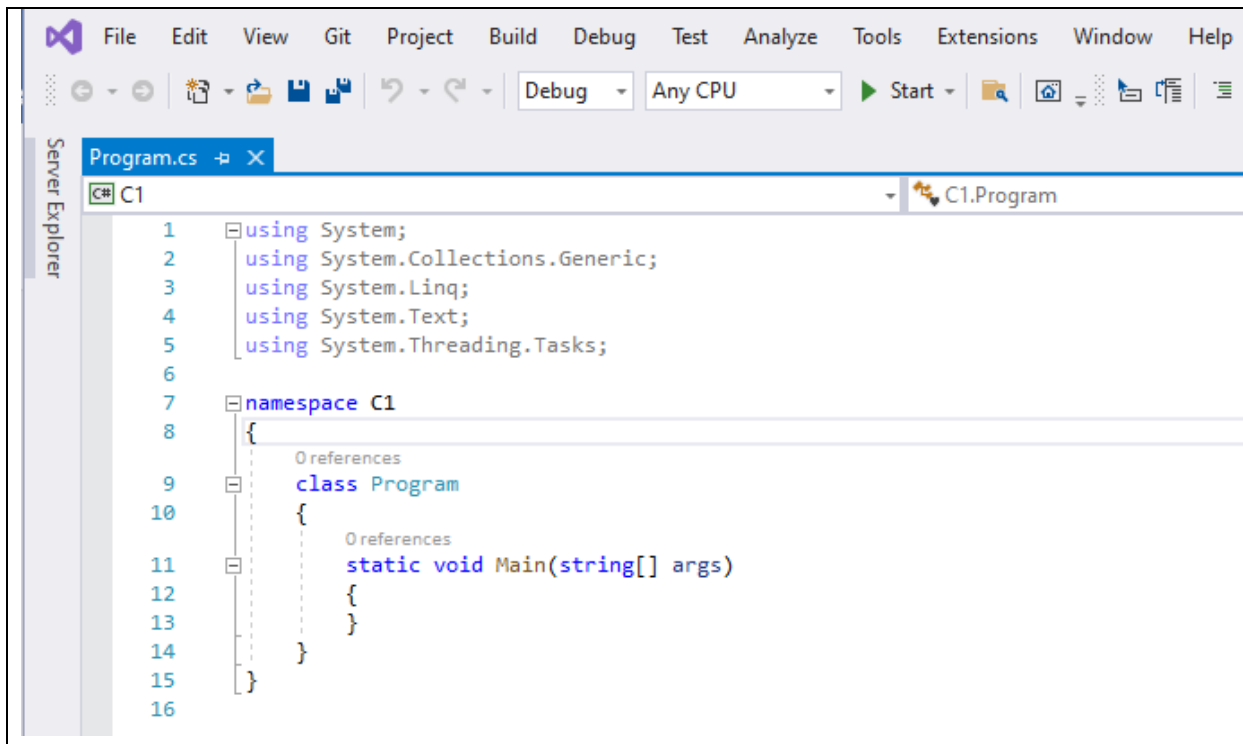


Рисунок 1.5– Вкладка програмного коду

Після рядка `static void Main()` починаємо вводити програмний код.

Програмний код:

```
using System; // Підключення простору імен System
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace C1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(" Ми вивчаємо мову програмування C# ");
            Console.ReadLine();
        }
    }
}
```

3. Викличте контекстне меню редактора, натиснувши правою кнопкою миші на порожньому місці у вікні редагування коду. Виконайте команду *Remove and Sort Using* (Видалити та відсортувати директиви *using*). Нижче наведено програму після виконання команди.

```
using System; // Підключення простору імен System

namespace C1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(" Ми вивчаємо мову програмування C# ");
            Console.ReadLine();
        }
    }
}
```

4. Запустіть програму командою
Debug/ Start Debugging (Налагоджування/ Почати налагоджування) <F5>

або натисніть на зелену стрілку ► *Start* на панелі інструментів
Результати розрахунку подано на рис. 1.6.

Якщо немає помилок, створюється єдиний завантажений файл із розширенням *.exe*.



Рисунок 1.6 – Результати розв’язання задачі

5. Зберегти проект
File/ Save All (Файл/ Зберегти все)

Проект у Visual Studio складається з файлу проекту (файл з розширенням *.csproj*), файлу розв’язання (з розширенням *.sln*), файла початкового тексту (з розширенням *.cs*) та ін. На диску D:\ Program з’явилась тека з ім’ям C1, що містить зазначені файли.

Файл розв’язання знаходиться в теці
d:\ Program \C1\ C1.sln

Програма, що відкомпілювалася, знаходиться в теці
d:\Program\C1\C1\bin\Debug\C1.exe

Для відкриття існуючого проекту використовується команда *Open a project or solution*. У вікні *Open project / solution* відкрити теку з ім'ям проекту, виділити файл з розширенням *.sln*, натиснути кнопку *Open* або зробити подвійне клацання на файлі з розширенням *.sln*

Пояснення до програми:

1) `using System;`

На початку програми розташовується розділ підключення просторів імен, кожен рядок якого починається ключовим словом `using`. У цьому розділі описуються використовувані простори імен. Кожен простір імен включає класи, що виконують певну роботу.

Рядок `using System;` означає, що в програмі використовується простір імен `System`. У `C#` простір імен визначає ділянку оголошень. Ми вимушені використати простори імен тих бібліотек, засоби яких застосовуються в наших програмах. Простір імен `System` об'єднує ті класи з бібліотеки класів середовища `.NET Framework`, які найчастіше використовуються в консольних програмах на `C#`.

Застосування в програмі бібліотеки класів передбачає або створення об'єктів класів цієї бібліотеки, або звернення до статичних полів і методів бібліотечних класів. З простору імен `System` у нашій програмі використовується клас `Console` і два статичні методи цього класу `WriteLine()` і `ReadLine()`. Помістивши в програму оператор `using System`, можна звертатися до названих методів за допомогою скорочених складених імен `Console.WriteLine()` і `Console.ReadLine()`. Якщо цей рядок не писати, тоді при записі в програмі, наприклад методу `Console.WriteLine()`, потрібно буде вказувати простір імен, до якого цей метод належить, тобто `System.Console.WriteLine()`. Вказувати простір імен `System` щоразу, коли використовується член цього простору, стомливе заняття, і тому більшість програмістів на `C#` вводять директиву `using System` на початку своїх програм.

2) `// Підключення простору імен System`

Це однорядковий коментар, який розпочинається з символів `//`, зона дії коментаря поширюється до кінця рядка і використовується для стислих пояснень.

3) namespace C1

Це оголошення вводить для програми власний простір імен з позначкою C1. Це ім'я ми визначили при створенні проекту. Програма містить створений автоматично клас (Program), а в складі цього класу – метод Main().

4) class Program

Відповідно до об'єктної орієнтації мови C#, усяка програма, написана мовою C#, є класом або сукупністю класів. У нашій програмі один клас з ім'ям Program. Ім'я класу можна вибрати довільно. Опис класу розпочинається з ключового слова class, за яким йде його ім'я і далі у фігурних дужках – тіло класу – список елементів класу (його даних і функцій, що називаються також методами).

У класі Program тільки один метод із заголовком static void Main().

5) static void Main()

Це статичний метод зі спеціальним ім'ям Main. Цей метод визначає точку входу в програму – саме з виконання операторів методу Main() починається виконання її коду. Зазвичай програма складається не з одного методу, але Main (головний) присутній завжди.

Ключове слово void вказує на те, що метод Main() не повертає значення. Порожні круглі дужки після імені методу Main означають, що цим методом не передається ніякої інформації. Весь код, що становить тіло методу, знаходиться між відкриваючими і закриваючими фігурними дужками. У середині методу розташовуються дії, які цей метод виконує.

```
6) Console.WriteLine(" Ми вивчаємо мову програмування C#");
```

На консоль виводиться рядок " Ми вивчаємо мову програмування C# ".

7) Console.ReadLine();

Оператор Console.ReadLine() використовується для організації очікування натискання клавіші <Enter> після завершення консольної програми (при налагоджуванні, щоб вікно результатів не закривалося відразу після завершення програми).

Задача 1.2. Задано довжини двох катетів прямокутного трикутника a, b . Обчислити довжину гіпотенузи c і величини двох його кутів α і β (рис. 1.7).

Значення a, b увести з клавіатури.

Гіпотенуза c обчислюється за формулою:

$$c = \sqrt{a^2 + b^2},$$

кути трикутника α і β в радіанах розраховуються так:

$$\alpha = \arctg\left(\frac{a}{b}\right), \quad \beta = \arctg\left(\frac{b}{a}\right),$$

кути трикутника α і β в градусах розраховуються за формулами:

$$\alpha_{\text{гр}} = \alpha \cdot 180/\pi; \quad \beta_{\text{гр}} = \beta \cdot 180/\pi$$

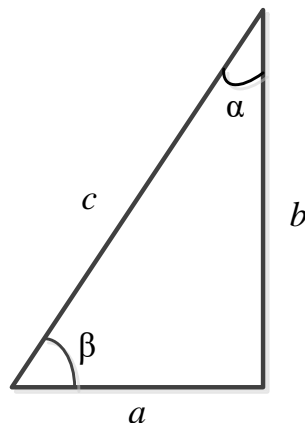


Рисунок 1.7 – Прямокутний трикутник до задачі 1.2

Розв’язання задачі можна розбити на такі етапи:

1. Уведення величин a і b з клавіатури в пам’ять комп’ютера.
2. Розрахування значень c, α і β за наведеними вище формулами.
3. Виведення значень c, α і β на екран дисплея.

Блок-схему розв’язання задачі подано на рис. 1.8

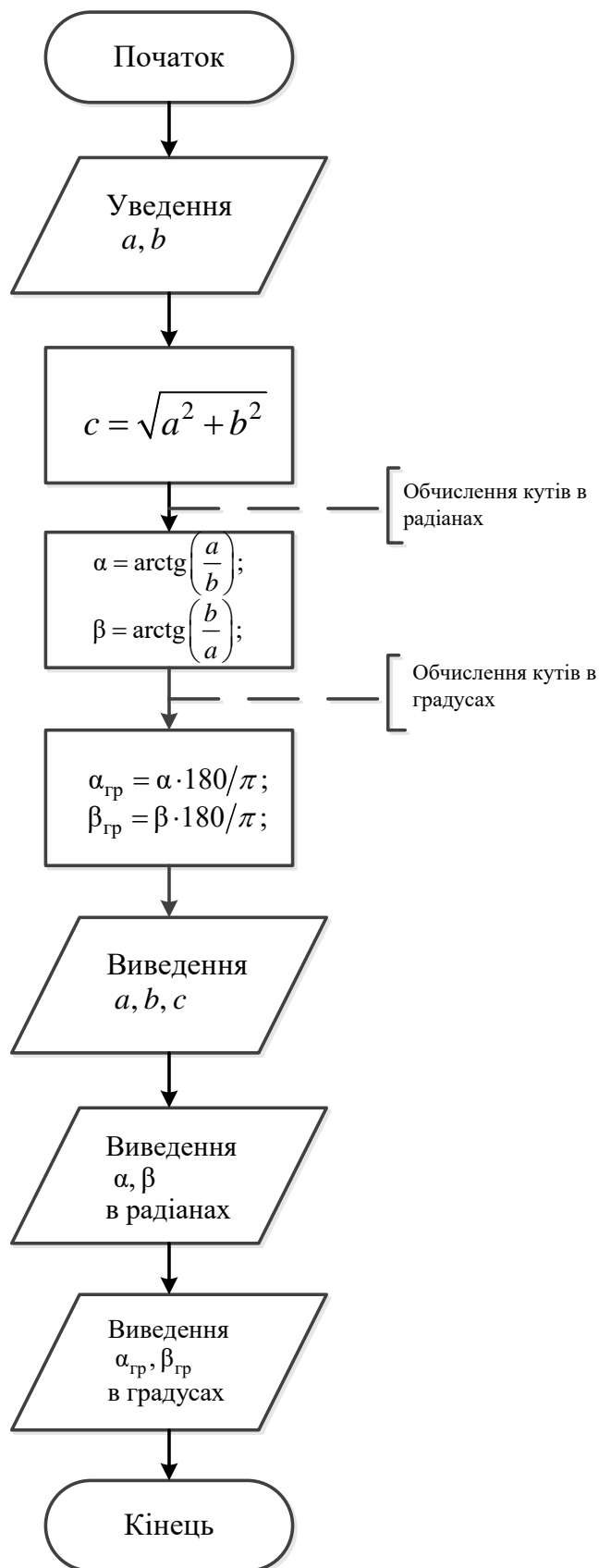


Рисунок 1.8 – Блок-схема алгоритму до задачі 1.2

Вхідні дані подамо як змінні, їх значення будуть вводитися з клавіатури, результати виводяться на екран монітора в консольне вікно.

Програмний код:

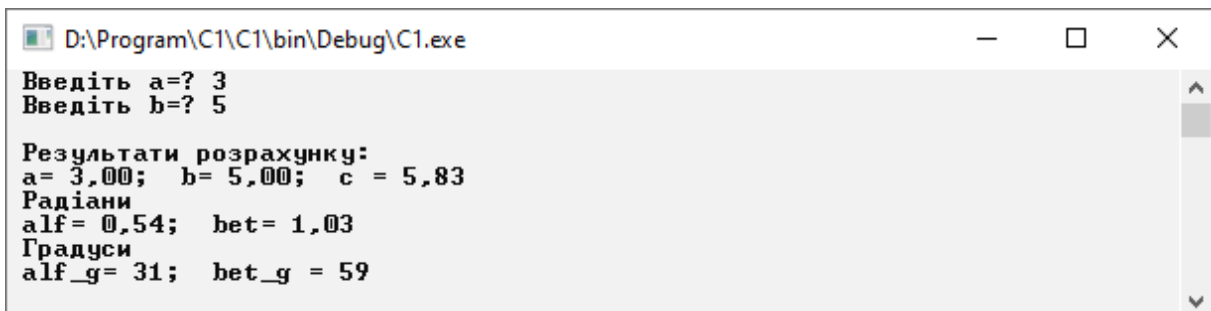
```
using System;

namespace C1
{
    class Program
    {
        static void Main(string[] args)
        {
            // Перша програма мовою C#
            double a, b, c, alf, bet, alf_g, bet_g;
            Console.Write(" Введіть a=? ");
            a = Convert.ToDouble(Console.ReadLine());
            Console.Write(" Введіть b=? ");
            b = Convert.ToDouble(Console.ReadLine());
            c = Math.Sqrt(a * a + b * b);
            alf = Math.Atan(a / b);
            bet = Math.Atan(b / a);
            alf_g = alf * 180 / Math.PI;
            bet_g = bet * 180 / Math.PI;
            Console.WriteLine("\n Результати розрахунку:");
            Console.WriteLine($" a= {a:F2}; b= {b:F2};
                                c= {c:F2}");

            Console.WriteLine(" Радіани ");
            Console.WriteLine($" alf= {alf:F2}; bet= {bet:F2}");
            Console.WriteLine(" Градуси ");
            Console.WriteLine($" alf_g= {alf_g:F0};
                                bet_g= {bet_g:F0}");

            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 1.9.



```
D:\Program\C1\C1\bin\Debug\C1.exe
Введіть a=? 3
Введіть b=? 5

Результати розрахунку:
a= 3,00; b= 5,00; c = 5,83
Радіани
alf= 0,54; bet= 1,03
Градуси
alf_g= 31; bet_g = 59
```

Рисунок 1.9 – Результати розв’язання задачі 1.2

Пояснення до програми:

1) Розміщення операторів на рядках

Програма складається з операторів. Оператори закінчуються крапкою з комою, яка не ставиться перед відкриваючою і після закриваючою фігурними дужками. Фігурні дужки використовуються для створення програмних блоків, що дозволяють виконувати послідовність операторів як єдине ціле. В один рядок можуть бути записані декілька операторів, а один оператор може бути записаний в декілька рядків (розрив проводять по пропусках, арифметичних операторах).

2) //Перша програма мовою C#

Це однорядковий коментар, який розпочинається з символів //, зона дії коментаря поширюється до кінця рядка і використовується для стислих пояснень.

Багаторядкові коментарі обмежуються символами / * * /. Приклад запису багаторядкового коментаря:

```
/ * Програма для роботи з файлами  
    послідовного доступу * /
```

3) double a, b, c, alf, bet, alf_g, bet_g;

Оголошуються змінні a, b, c, alf, bet, alf_g, bet_g; – дійсні (дробові) числа подвійної точності типу double.

Змінні мають бути оголошені до їх застосування. Для оголошення змінної (змінних) спочатку вказується тип, а потім ім'я (імена, розділені комою).

Для дійсних змінних частіше використовується тип double, для цілих – тип int. При оголошенні змінної (зазначенні імені і типу) під цю змінну буде зарезервована ділянка комп'ютерної пам'яті відповідного розміру. У цій ділянці і зберігатиметься числове значення змінної.

Ім'я змінної (ідентифікатора) складається з латинських літер, цифр і символу підкреслення. Ім'я не може розпочинатися з цифри. У мові C# великі і малі букви розрізняються.

4) Console.WriteLine(" Введіть a= ");

Метод виводить на консоль (екран) рядок символів
Введіть a=

Курсор на новий рядок не переводиться.

```
5) a = Convert.ToDouble(Console.ReadLine());
```

Оператор присвоювання. Змінній `a` присвоїти значення, що вводить з консолі (клавіатури).

Користувач у відповідь на запрошення «Введіть `a=` » вводить з клавіатури число, значення змінної `a`.

Метод `Console.ReadLine()` вводить рядок символів. Функція `Convert.ToDouble` конвертує (перетворює) текст на числовий тип подвійної точності. Коли в програмі зустрічається оператор `ReadLine()`, його дія призупиняється доти, поки не буде уведено початкове дане.

```
6) c = Math.Sqrt(a * a + b * b);
```

```
7) alf = Math.Atan(a / b);
```

```
8) bet = Math.Atan(b/a);
```

Оператори присвоювання. Математичні функції C# є методами класу `Math`. Їх можна побачити, набравши `Math` і поставивши `(.)`. У випадковому списку ви побачите безліч математичних функцій : `abs`, `sin`, `cos`. та ін. і дві константи `E=2,71` (основа натуральних логарифмів) і `PI=3,14` (кількість діаметрів, укладених уздовж кола).

Для обчислення x^y в мові C# використовується функція `Math.Pow(x, y)`

Результат обчислення математичної функції – тип `double`, аргументи функцій також повинні бути типу `double`, наприклад, `sin(double x)`. При використанні тригонометричних функцій необхідно враховувати, що в мові C# кути обчислюються в радіанах.

```
9) Console.WriteLine("\n результати розрахунку:");
```

На екран виводиться текст «Результати розрахунку :», заздалегідь додається порожній рядок.

`\n` – керуюча послідовність символів – новий рядок

```
10) Console.WriteLine($"a={a:F2}; b={b:F2}; c={c:F2}");
```

Для виконання операцій виведення на консоль використовуються методи `Console.WriteLine()` і `Console.Write()`. Обидва методи виводять рядок тексту, але `Console.Write` залишає курсор у позиції закінчення рядка виведення, а `Console.WriteLine` крім цього додає операції переведення рядка і повернення каретки, тобто переміщає курсор у початок наступного рядка. Метод `Console.WriteLine()` з порожнім списком параметрів додає порожній рядок.

В даному прикладі знак \$ перед рядком вказує, що буде здійснюватися інтерполяція рядків. За допомогою інтерполяції рядків в C# можна вставити значення в один результуючий рядок. У середині рядка використовуються {...} фігурні дужки, всередині фігурних дужок можна вказувати не тільки змінні, але і різні вирази мови C#, які ми хочемо вивести. Інтерполяція по суті являє більш лаконічне форматування.

Текст, який не внесено в фігурні дужки, виводиться як є, замість фігурних дужок підставляються значення в зазначеному форматі.

{a:F2}; позначає наступне: при виведенні значення змінної a, використовувати формат F (Fixed) – число з фіксованою точкою, два знака після коми. При необхідності можна вказувати і ширину поля, яке відводиться під виведене значення, наприклад, формат {a,9:F2} позначає наступне: вивести значення змінної a, ширина поля – 9 знаків, використовувати формат з фіксованою комою, з точністю два знаки після коми. Якщо виведеному числу досить меншої кількості позицій, невикористовувані позиції заповнюються пробілами. Якщо числу потрібно більше позицій, параметр ігнорується.

Часто застосовується науковий формат E, наприклад, {a:E5} означає виведення змінної a в науковому форматі (E), з точністю 5 знаків після коми.

```
11) Console.WriteLine($"alf={alf:F2};   bet={bet:F2}");
```

Виведення значень кутів α і β у радіанах.

```
12) Console.WriteLine($"alf_g={alfg_:F0};  
                        bet_g={bet_g:F0}");
```

Виведення значень кутів α і β у градусах.

```
13) Console.ReadLine();
```

Оператор Console.ReadLine() використовується для організації очікування натискання клавіші <Enter> після завершення консольної програми (при налагоджуванні, щоб вікно результатів не закривалося відразу після завершення програми).

Задача 1.3. Увести два числа, виконати нескладні обчислення і вивести результат (застосувати другий спосіб виведення за форматом).

Програмний код:

```
using System;  
  
namespace C152  
{
```

```

class Program
{
    static void Main(string[] args)
    {
        int i;
        double x, y, z;
        Console.Write(" Введіть i=? "); // підказка при уведенні
        // Уведення, поєднане з перетворенням,
        // вводиться рядок, який перетвориться в ціле число
        i = Convert.ToInt32(Console.ReadLine());
        Console.Write(" Введіть x=? ");
        // Уводиться рядок, який перетвориться в дійсне число
        x = Convert.ToDouble(Console.ReadLine());
        y = i * x;
        z = Math.Pow((2 * i - x), 3);
        Console.WriteLine(" y=" + y.ToString("F3") + " z=" +
                           z.ToString("F4"));

        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 1.10.

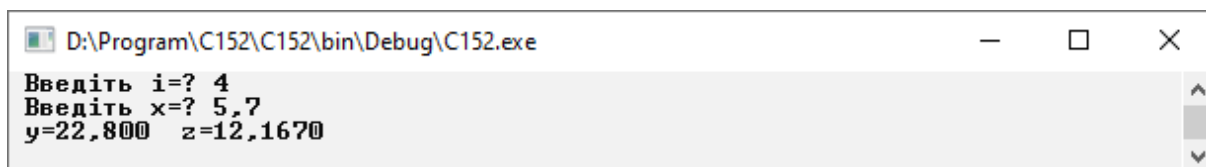


Рисунок 1.10 – Результати розв’язання задачі 1.3

Пояснення до програми:

```

Console.WriteLine(" y=" + y.ToString("F3") + " z=" +
                  z.ToString("F4"));

```

В цій задачі використовується *другий спосіб виведення за форматом* – вказати як параметр функції ToString символну константу, яка задає формат рядка-результату. Значення y виводиться з трьома знаками після коми, значення z – з чотирма знаками після коми. Символьна константа "F3" означає формат F(Fixed) – число з фіксованою крапкою, три знаки після коми.

У нашому прикладі метод Console.WriteLine послідовно виводить на екран текст y= і значення змінної y, потім текст z= і значення змінної z. При виведенні числа необхідно перетворити його в послідовність символів, тобто в рядок. Для цього служить метод ToString(). У цьому випадку він виконується явно, але можна викликати його і неявно:

```

Console.WriteLine(" y=" + y + "z=" + z);

```

Якщо аргумент методу WriteLine() містить символьний рядок і число, то виконується автоматичне перетворення числа в рядок. Операція (+) – зчеплення (конкатенація) рядків.

Задача 1.4. Обчислити значення виразів.

$$y = b \operatorname{tg}^2 x - \frac{a}{\sin^2(x/a)};$$

$$d = a e^{-\sqrt{a}} \cos(bx/a)$$

за умови $a = 3,2$; $b = 17,5$; $x = -4,8$.

Спосіб 1. Початкові дані подано як константи, їх значення уведені в програмному кодї, результати виводяться на консоль.

Блок-схему розв'язання задачі наведено на рис. 1.11.

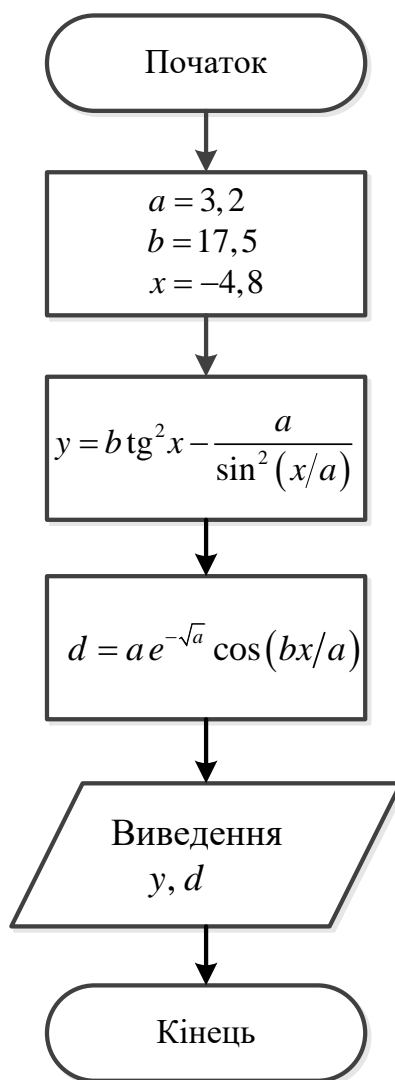


Рисунок 1.11 – Блок-схема алгоритму до задачі 1.4 (спосіб 1, 2)

Програмний код:

```
using System;

namespace K1
{
    class Program
    {
        static void Main(string[] args)
        {
            // Програма 1
            const double a = 3.2, b = 17.5, x = -4.8;
            double y, d;
            y = b * Math.Pow(Math.Tan(x), 2) -
                a / Math.Pow(Math.Sin(x / a), 2);
            d = a * Math.Exp(-Math.Sqrt(a)) * Math.Cos(b * x / a);
            Console.WriteLine($" y= {y:F3}; d= {d:F4}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 1.12.



Рисунок 1.12 – Результати розв’язання задачі 1.4 (спосіб 1)

Спосіб 2. Початкові дані подано як змінні, їх значення уведені в програмному коді, результати виводяться на консоль.

Блок-схему розв’язання задачі наведено на рис. 1.11.

Програмний код:

```
using System;

namespace K2
{
    class Program
    {
        static void Main(string[] args)
        {
            // Програма 2
            double a, b, x, y, d;
            a = 3.2; b = 17.5; x = -4.8;
            y = b * Math.Pow(Math.Tan(x), 2) -
                a / Math.Pow(Math.Sin(x / a), 2);
            d = a * Math.Exp(-Math.Sqrt(a)) * Math.Cos(b * x / a);
        }
    }
}
```

```

        Console.WriteLine($" y= {y:F3}; d= {d:F4}");
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 1.13.



Рисунок 1.13 – Результати розв’язання задачі 1.4 (спосіб 2)

Пояснення до програми:

1) $a = 3.2$; $b = 17.5$; $x = -4.8$;

Значення початковим даним задаємо за допомогою операторів присвоювання. У коді програми при записі дійсних чисел десятковим розділювачем $\epsilon < . >$, але при роботі програми в операціях уведення/ виведення використовується десятковий розділювач, вказаний у налаштуваннях системи $< , >$.

Спосіб 3. Значення вхідних даних уводяться з консолі (клавіатура), результати виводяться на консоль (екран монітора).

Блок-схему розв’язання задачі подано на рис. 1.14.

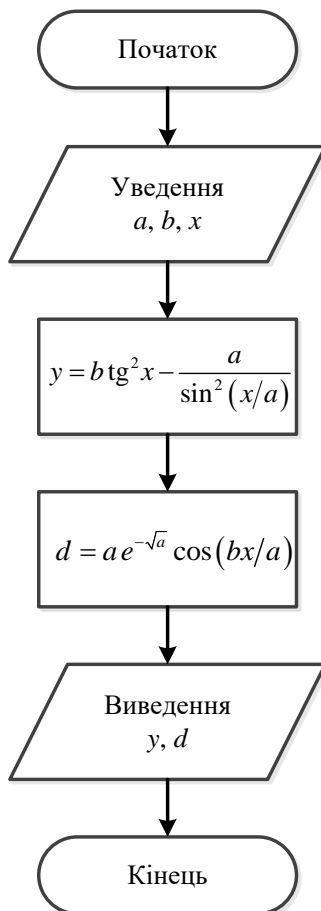


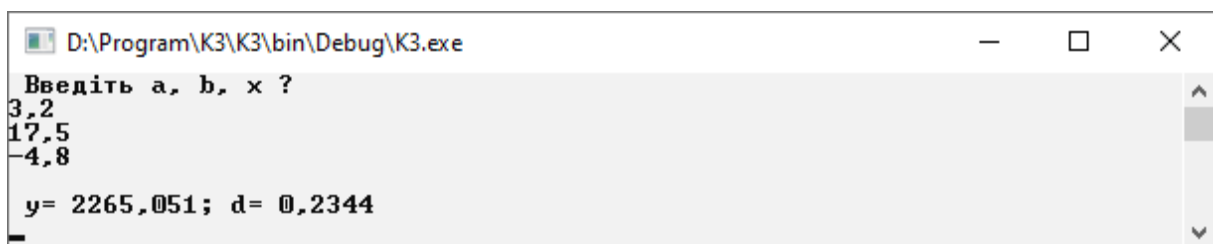
Рисунок 1.14 – Блок-схема алгоритму до задачі 1.4 (спосіб 3)

Програмний код:

```
using System;

namespace K3
{
    class Program
    {
        static void Main(string[] args)
        {
            //Програма 3
            double a, b, x, y, d;
            Console.WriteLine(" Введіть a, b, x ? ");
            a = Convert.ToDouble(Console.ReadLine());
            b = Convert.ToDouble(Console.ReadLine());
            x = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            y = b * Math.Pow(Math.Tan(x), 2) -
                a / Math.Pow(Math.Sin(x / a), 2);
            d = a * Math.Exp(-Math.Sqrt(a)) * Math.Cos(b * x / a);
            Console.WriteLine($" y= {y:F3}; d= {d:F4}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 1.15.



```
D:\Program\K3\K3\bin\Debug\K3.exe
Введіть a, b, x ?
3,2
17,5
-4,8
y= 2265,051; d= 0,2344
```

Рисунок 1.15 – Результати розв’язання задачі 1.4 (спосіб 3)

Пояснення до програми:

1) `Console.WriteLine(" Введіть a, b, x: ");`

Метод виводить на консоль (екран) рядок символів

Введіть a, b, x:

Курсор переводиться на новий рядок.

2) `a = Convert.ToDouble(Console.ReadLine());`
`b = Convert.ToDouble(Console.ReadLine());`
`x = Convert.ToDouble(Console.ReadLine());`

Оператори присвоювання. Змінним a , b , x присвоїти значення, що вводиться з консолі (клавіатури).

Користувач у відповідь на запрошення «Введіть a , b , x :» вводить з клавіатури числа : значення змінних, закінчуючи уведення кожного числа натисненням клавіші <Enter>.

Метод `Console.ReadLine()` вводить рядок символів.

Функція `Convert.ToDouble` конвертує (перетворює) текст у числовий тип подвійної точності.

Коли в програмі зустрічається оператор `ReadLine()`, її дія призупиняється доти, доки не буде уведено початкове дане.

1.2 Завдання для виконання лабораторних робіт (частина 1)

Таблиця 1.1 – Варіанти задач

Номер вар.	Розрахункові формули	Початкові дані
1	2	3
1	$a = \frac{2\cos(x - \pi/6)}{1/2 + \sin^2 y}$ $b = 1 + \frac{z^2}{3 + z^2/5}$	$x = 1,426$ $y = -1,22$ $z = 3,5$
2	$\gamma = x^{y/x} - \sqrt[3]{y/x} $ $\psi = (y - x) \frac{(y - z)}{1 + (y - x)^2}$	$x = 1,825$ $y = 18,225$ $z = -3,298$
3	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }$ $s = b \sin(at^2 \cos 2t) - 1$	$a = -0,5$ $b = 1,7$ $t = 0,44$
4	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x + a)/x$ $y = \cos^2 x - x/\sqrt{a^2 + b^2}$	$a = 1,5$ $b = 15,5$ $x = -2,9$
5	$s = x^3 \operatorname{tg}(x + b)^2 + a/\sqrt{x + b}$ $Q = \frac{bx^2 - a}{e^{ax} - 1}$	$a = 16,5$ $b = 3,4$ $x = 0,61$

Закінчення табл. 1.1

1	2	3
6	$R = x^2(x+1)/b - \sin^2(x+a)$ $S = \sqrt{xb/a} + \cos^2(x+b)$	$a = 0,7$ $b = 0,05$ $x = 0,5$
7	$y = \sin^3(x^2 + a) - \sqrt{x/b}$ $z = \frac{x^2}{a} + \cos(x+b)^3$	$a = 1,1$ $b = 0,004$ $x = 0,2$
8	$f = \sqrt[3]{m \operatorname{tg} t + c \sin t }$ $z = m \cos(bt \sin t) + c$	$m = 2; \quad c = -1$ $t = 1,2; \quad b = 0,7$
9	$y = \ln^3(1+x^2)$ $F = \sin x^2 \cos \frac{7x-2}{3,75\pi}$	$x = 1,45$
10	$f = \ln(a+x^2) + \sin^2(x/b)$ $z = e^{-cx} \frac{x + \sqrt{x+a}}{x - \sqrt{ x-b }}$	$a = 10,2; \quad b = 9,2$ $x = 2,2; \quad c = 0,5$

1.3 Завдання для виконання лабораторних робіт (частина 2)

Задача 1. Написати програму обчислення площі трикутника, якщо відомі довжини двох його сторін і величина кута між цими сторонами (кут заданий у градусах):

$$S = \frac{1}{2} a \cdot b \cdot \sin \alpha .$$

Пояснення: Аргумент вбудованої математичної функції `Math.Sin(x)` повинен бути в радіанах. Для переведення кутів з градусів у радіани необхідно величину кута в градусах помножити на дріб $\frac{\pi}{180}$. При запису дробу $1/2$ маємо цілочислове ділення, при якому остача відкидається. Щоб цього не сталося, міняємо тип хоча б одного з операндів: `1.0/2.0` або `(double) 1/2`.

Нижче наведено рекомендований вигляд екрана програми

Обчислення площі трикутника

Уведіть довжину сторони трикутника a =? **25**

Уведіть довжину сторони трикутника b =? **17**

Уведіть величину кута між сторонами трикутника =? **30**

Площа трикутника: 106.25 кв. см.

Задача 2. Написати програму обчислення об'єму і площі поверхні циліндра

$$V = \pi r^2 h; \quad S = 2\pi r^2 + 2\pi r h.$$

Нижче наведено рекомендований вигляд екрана програми (дані, введені користувачем, виділено напівжирним).

Обчислення об'єму циліндра

Уведіть початкові дані:

Радіус основи (см) =? **5,5**

Висота циліндра (см) =? **7**

Об'єм циліндра **665,23** см.куб.

Площа поверхні циліндра: **431,97** кв. см.

Таблиця 1.2 – Арифметичні операції

Операція	Знак	Запис	Дія
Арифметичне додавання	+	a + b	Додає два числа
Унарний плюс	+	+a	
Арифметичне віднімання	-	a - b	Віднімає з одного числа інше
Унарний мінус	-	-a	
Множення	*	a * b	Помножає два числа
Ділення	/	a / b	Ділить два числа і повертає результат з рухомою точкою
Ділення по модулю (операція обчислення остачі)	%	a%b	Отримання остачі від цілочислового ділення (7%3=1). Можна застосовувати і до типів з рухомою точкою (10.0%3.0=1)
Присвоєння	=	a=b	Задає нове значення змінній
Інкремент	++	i++; ++i;	постфіксна форма, i = i+1; префіксна форма, i = i+1
Декремент	--	i--; --i;	постфіксна форма, i = i-1; префіксна форма, i = i-1
Доступ до елемента	. (точка)		Вибір члена (класа або об'єкта)
Виділення пам'яті	new		Створення об'єкта (створення екземпляра)

Таблиця 1.3 – Список математичних методів класу System.Math

Математичні методи	Позначення в математиці	Призначення
1	2	3
Math.Abs(x)	$ x $	Модуль (абсолютна величина числа x)
Math.Acos(x)	$\arccos x$	Арккосинус числа x (в діапазоні від -1 до 1)
Math.Asin(x)	$\arcsin x$	Арксинус числа x (в діапазоні від -1 до 1)
Math.Atan(x)	$\arctg x$	Арктангенс числа x (в діапазоні від -1 до 1)
Math.Ceiling(x)		Округлення до більшого цілого. Повернення найближчого цілого, більшого, ніж значення аргументу x <i>Приклад</i> Math.Ceiling(1.04) дасть у результаті 2
Math.Cos(x)	$\cos x$	Косинус кута x . Кут x задається в радіанах
Math.E	e	Основа натурального логарифму ($e = 2,718$)
Math.Exp(x)	e^x	Експонента. e – основа натурального логарифму, піднесена до степеня x
Math.Floor(x)		Округлення до меншого цілого. Повертає найближче ціле число, менше, ніж значення аргументу x <i>Приклад</i> Math.Floor(1.04) дасть у результаті 1
Math.IEEERemainder(x,y)		Остача від ділення. Повертає остачу від ділення числа x на число y
Math.Log(x)	$\ln x$	Натуральний логарифм x , де $x > 0$
Math.Log10(x)	$\lg x$	Десятковий логарифм x
Math.Log(x, a)	$\log_a x$	Логарифм x по основі a
Math.Max(x,y)		Максимум з двох чисел
Math.Min(x,y)		Мінімум з двох чисел
Math.PI – константа		Значення числа $\pi = 3,1415926$
Math.Pow(x,y)	x^y	Піднесення x до степеня y
Math.Round(x)		Округлення. Округляє задане число до найближчого цілого (арифметичне округлення) <i>Приклад</i> Math.Round(3.1) дає в результаті 3. Math.Round(3.8) дає в результаті 4
Math.Round(x, n)		Округлення. Округляє число з рухомою точкою до заданої кількості дробових розрядів n

Закінчення табл. 1.3

1	2	3
Math.Sign(x)		Знак числа. Повертає значення, що визначає знак цілого числа. $\text{Sign}(x) = \begin{cases} -1, & x < 0 \\ 0, & \text{якщо } x = 0 \\ +1, & x > 0 \end{cases}$
Math.Sin(x)	sin x	Синус кута x. Кут x задається в радіанах
Math.Sqrt(x)	\sqrt{x}	Корінь квадратний з числа x, де $x \geq 0$
Math.Tan(x)	tg x	Тангенс кута x. Кут x задається в радіанах
Math.Truncate(x)		Обчислює цілу частину заданого десяткового числа
Rnd(x)		Випадкове число з діапазона 0 – 1

Примітка. Результат обчислення більшості математичних функцій – тип double, аргументи функцій також повинні бути типу double, наприклад, Math.Sin(double x). Назви функцій пишуться з великої літери. Результат функції Math.Sign(x) має тип int.

Таблиця 1.4 – Оператори порівняння

Оператор	Знак	Запис
дорівнює	==	a==b
не дорівнює	!=	a!=b
більше	>	a>b
менше	<	a<b
більше або дорівнює	>=	a>=b
менше або дорівнює	<=	a<=b

Таблиця 1.5 – Логічні оператори

Оператор	Значення
&	логічне І (AND)
	логічне АБО (OR)
^	логічне виключне АБО (XOR)
!	логічне заперечення з НЕ (NOT)
&&	умовне І (AND)
	умовне АБО (OR)

Пріоритет виконання операторів:

- 1) унарні операції (операції з одним операндом, наприклад зміна знака);
 - 2) оператори множення / ділення;
 - 3) оператори додавання / віднімання;
 - 4) оператори зсуву;
 - 5) оператори порівняння (причому оператори дорівнює / не дорівнює мають менший пріоритет);
 - 6) логічні;
 - 7) оператор присвоєння виконується останнім.
- Оператори одного рівня пріоритету виконуються зліва направо.
Порядок обчислень за замовчуванням можна змінити за допомогою круглих дужок.

Тема 2. СКЛАДАННЯ ПРОГРАМ РОЗГАЛУЖЕНОЇ СТРУКТУРИ

Мета роботи: вивчити оператори розгалуження if i switch.

2.1. Приклади розв'язання задач

Задача 2.1. Обчислити значення функції

$$y = \begin{cases} x^2 + 5/a, & \text{якщо } x < 0 \\ 12 + a, & \text{якщо } x \geq 0 \end{cases}$$

за умови $a = 5,2$; $x = -3$; 0 ; $34,85$.

Блок-схему розв'язання задачі подано на рис. 2.1.

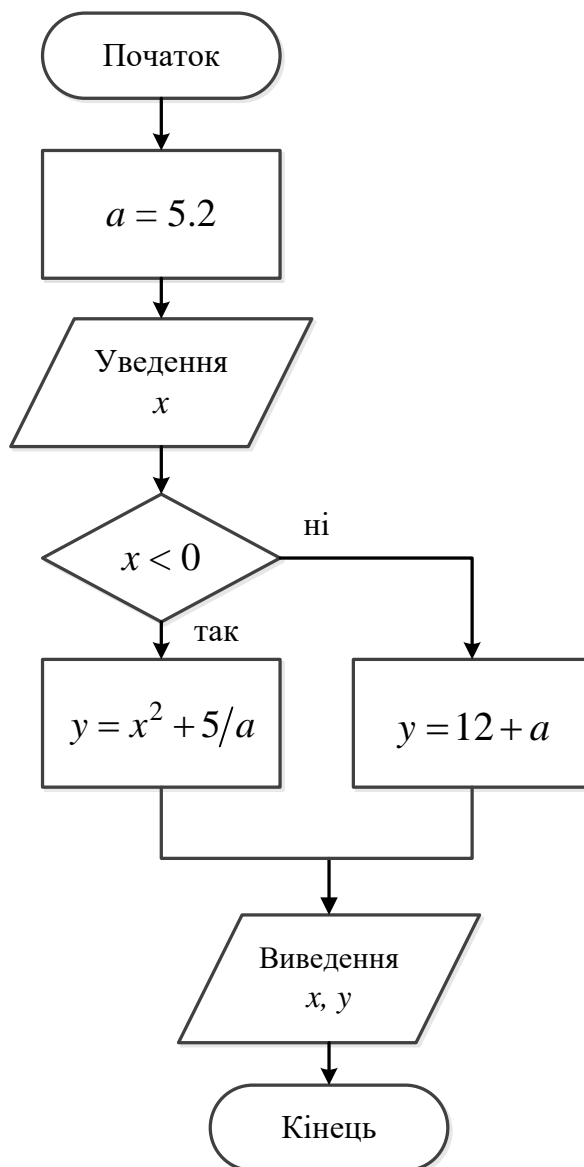


Рисунок 2.1 – Блок-схема алгоритму до задачі 2.1

Програмний код:

```
using System;

namespace C4
{
    class Program
    {
        static void Main(string[] args)
        {
            double a, x, y;
            a = 5.2;
            Console.Write(" Введіть x=? ");
            x = Convert.ToDouble(Console.ReadLine());
            if (x < 0)
                y = x * x + 5 / a;
            else
                y = 12 + a;
            Console.WriteLine($" x= {x:F2}; y= {y:F3}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку наведено на рис. 2.2.



Рисунок 2.2 – Результати розв’язання задачі 2.1

Задача 2.2. Написати програму розв’язання квадратного рівняння

$$ax^2 + bx + c = 0.$$

Можна виділити такі етапи розв’язання:

1. Увести числові значення змінних a, b, c .
2. Обчислити значення дискримінанта d за формулою $d = b^2 - 4ac$.

3. Якщо $d \geq 0$ та $a \neq 0$, тоді провести обчислення коренів

$$X_1 = \frac{-b + \sqrt{d}}{2a}, \quad X_2 = \frac{-b - \sqrt{d}}{2a} \text{ і виведення їх на екран.}$$

Перейти до пункту 4.

В інших випадках вивести повідомлення «Дійсних коренів немає» і перейти до пункту 4.

4. Припинити обчислення.

На третьому етапі розв'язання задачі скористаємося оператором *if* .

Початкові дані: дійсні числа a, b, c – коефіцієнти квадратного рівняння.

Результати роботи програми : дійсні числа X_1 і X_2 (корені квадратного рівняння) або повідомлення про те, що коренів немає.

Допоміжні змінні: дійсна змінна d для обчислення дискримінанта квадратного рівняння.

Блок-схему розв'язання задачі подано на рис. 2.3.

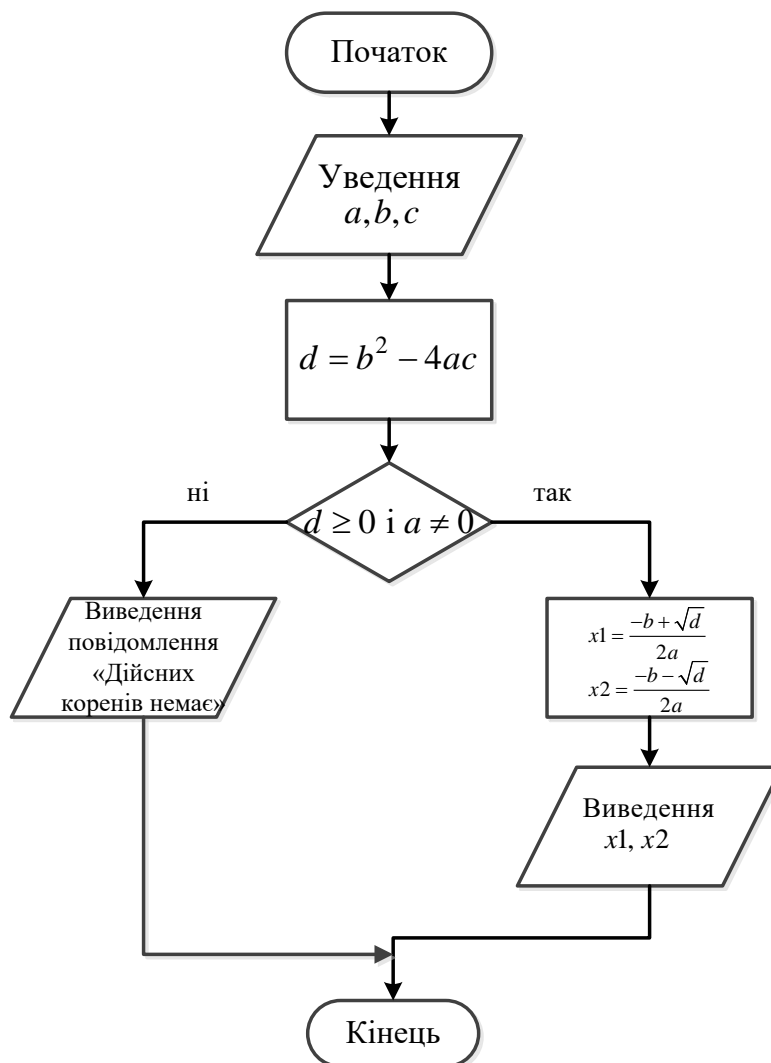


Рисунок 2.3 – Блок-схема алгоритму до задачі 2.2

Програмний код:

```
using System;

namespace C7a
{
    class Program
    {
        static void Main(string[] args)
        {
            double a, b, c, D, x1, x2;
            // Введення значень коефіцієнтів квадратного рівняння
            Console.Write(" Введіть a=? ");
            a = Convert.ToDouble(Console.ReadLine());
            Console.Write(" Введіть b=? ");
            b = Convert.ToDouble(Console.ReadLine());
            Console.Write(" Введіть c=? ");
            c = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            // Обчислення дискримінанта
            D = b * b - 4 * a * c;
            Console.WriteLine($" Дискримінант D= {D:F3}");
            if (D >= 0 && a != 0)
            {
                x1 = (-b + Math.Sqrt(D)) / (2 * a);
                x2 = (-b - Math.Sqrt(D)) / (2 * a);
                Console.WriteLine($" x1= {x1:F3};    x2= {x2:F3}");
            }
            else
            {
                Console.WriteLine(" Дійсних коренів немає ");
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 2.4.

```

D:\Program\C7a\C7a\bin\Debug\C7a.exe
Введіть a=? 3
Введіть b=? 8
Введіть c=? 2

Дискримінант D= 40,000
x1= -0,279;   x2= -2,387

```

```

D:\Program\C7a\C7a\bin\Debug\C7a.exe
Введіть a=? 1
Введіть b=? 4
Введіть c=? 4

Дискримінант D= 0,000
x1= -2,000;   x2= -2,000

```

```

D:\Program\C7a\C7a\bin\Debug\C7a.exe
Введіть a=? 12
Введіть b=? 10
Введіть c=? 20

Дискримінант D= -860,000
Дійсних коренів немає

```

Рисунок 2.4 – Результати розв’язання задачі 2.2

Задача 2.3. Обчислити значення функції

$$f = \begin{cases} 1 - \cos a - y, & y < 1 \\ (ay + b)/2, & \text{якщо } 1 \leq y \leq 4 \\ a + 1, & y > 4 \end{cases}$$

за умови $a = 0,2$; $b = 0,001$; $y = 0,5$; 3 ; 12 .

Створити консольний додаток з використанням вкладених операторів if. Блок-схему розв’язання задачі подано на рис. 2.6.

Програмний код:

```

using System;

namespace C5
{
    class Program
    {
        static void Main(string[] args)
        {
            double a, b, y, f;
            a = 0.2; b = 0.001;

```

```

Console.Write(" Введіть y=? ");
y = Convert.ToDouble(Console.ReadLine());
if (y < 1)
    f = 1 - Math.Cos(a) - y;
else if ((1 <= y) && (y <= 4))
    f = (a * y + b) / 2;
else
    f = a + 1;
Console.WriteLine($" y= {y:F2}; f= {f:F3}");
Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 2.5.



Рисунок 2.5 – Результати розв’язання задачі 2.3

Пояснення до програми:

Математичний вираз $1 \leq y \leq 4$ у мові програмування C# розділяється на два вирази, об’єднані оператором && (І):

$(1 \leq y) \ \&\& \ (y \leq 4)$.

Оскільки операції порівняння мають більш високий пріоритет, ніж логічні, то дужки можна не ставити:

$1 \leq y \ \&\& \ y \leq 4$.

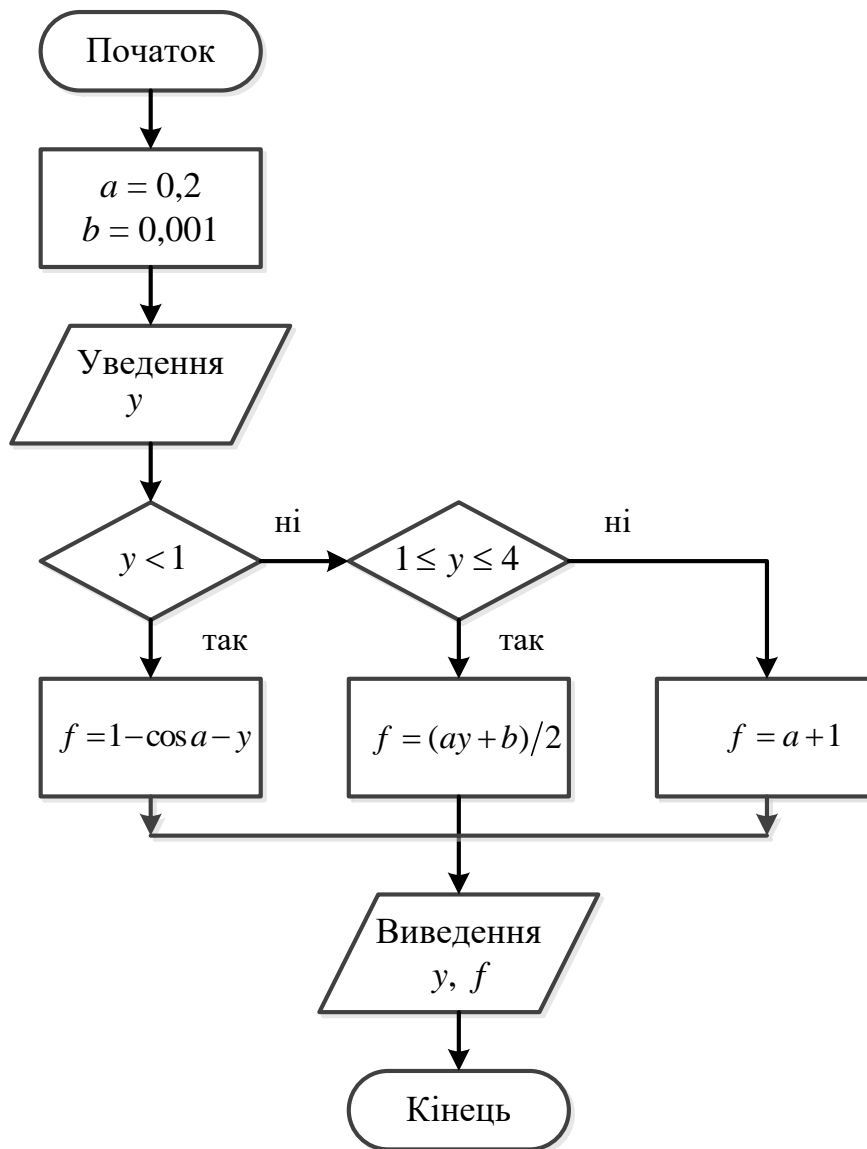


Рисунок 2.6 – Блок-схема алгоритму до задачі 2.3

Задача 2.4. Створіть консольний додаток для обчислення значення функції :

$$y = \begin{cases} (x \cdot z)^2, & x > 0, z > 0 \\ x^{-z}, & \text{якщо } x < 0, z < 0 \\ 5, & \text{в інших випадках} \end{cases}$$

за умови

$$x = 3; \quad z = 5$$

$$x = -5 \quad z = -4$$


$$x = 25 \quad z = -8$$

Програмний код:

```
using System;

namespace C6
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, z, y;
            Console.Write(" Введіть x=? ");
            x = Convert.ToDouble(Console.ReadLine());
            Console.Write(" Введіть z=? ");
            z = Convert.ToDouble(Console.ReadLine());
            if (x > 0 && z > 0)
                y = Math.Pow((x * z), 2);
            else if (x < 0 && z < 0)
                y = Math.Pow(x, -z);
            else
                y = 5;
            Console.WriteLine($" x={x};    z={z};    y={y}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 2.7.



The screenshot shows a console window titled "D:\Program\C6\C6\bin\Debug\C6.exe". The output text is: "Введіть x=? 3", "Введіть z=? 5", and "x=3; z=5; y=225".



The screenshot shows a console window titled "D:\Program\C6\C6\bin\Debug\C6.exe". The output text is: "Введіть x=? -5", "Введіть z=? -4", and "x=-5; z=-4; y=625".



The screenshot shows a console window titled "D:\Program\C6\C6\bin\Debug\C6.exe". The output text is: "Введіть x=? 25", "Введіть z=? -8", and "x=25; z=-8; y=5".

Рисунок 2.7– Результати розв’язання задачі 2.4

Задача 2.5. Обчислити значення функції

$$z = \begin{cases} 3a^2 - b, & a > 0, b > 0 \\ a + b, & \text{якщо } a < 0, b < 0 \\ a - 1, & a > 0, b < 0 \end{cases} \quad \text{за умови } \begin{cases} a = 2; & b = 5; \\ a = -3,5; & b = -7; \\ a = 5; & b = -8; \end{cases}$$

Програмний код:

```
using System;

namespace C135
{
    class Program
    {
        static void Main(string[] args)
        {
            double a, b, z;
            Console.Write(" Введіть a=? ");
            a = Convert.ToDouble(Console.ReadLine());
            Console.Write(" Введіть b=? ");
            b = Convert.ToDouble(Console.ReadLine());
            z = 0;
            if (a > 0 && b > 0)
                z = 3 * a * a - b;
            else if (a < 0 && b < 0)
                z = a + b;
            else if (a > 0 && b < 0)
                z = a - 1;
            Console.WriteLine($" a= {a}; b= {b}; z= {z:F2}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 2.8.

```
D:\Program\C135\C135\bin\Debug\C135.exe
Введіть a=? 2
Введіть b=? 5
a= 2; b= 5; z= 7,00
```

```
D:\Program\C135\C135\bin\Debug\C135.exe
Введіть a=? -3,5
Введіть b=? -7
a= -3,5; b= -7; z= -10,50
```

```
D:\Program\C135\C135\bin\Debug\C135.exe
Введіть a=? 5
Введіть b=? -8
a= 5; b= -8; z= 4,00
```

Рисунок 2.8 – Результати розв’язання задачі 2.5

Задача 2.6. Створіть консольний додаток для обчислення значення функції. Для розв’язання завдання використайте багатоступеневу конструкцію `if-else-if`, що складається з вкладених операторів `if`

$$y = \begin{cases} x, & x = 0 \\ x^3 + 6, & x = 1 \\ x^2 + 2, & \text{якщо } 2 \leq x \leq 7 \\ x/3, & x < 0 \text{ або } x > 100 \\ 5, & \text{в інших випадках} \end{cases}$$

Блок-схему розв’язання задачі подано на рис. 2.10.

Програмний код:

```
using System;

namespace C7
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y;
```

```

Console.Write(" Введіть x=? ");
x = Convert.ToDouble(Console.ReadLine());
if (x == 0)
    y = x;
else if (x == 1)
    y = x * x * x + 6;
else if (2 <= x && x <= 7)
    y = x * x + 2;
else if (x < 0 || x > 100)
    y = x / 3;
else
    y = 5;
Console.WriteLine($" x={x};    y={y}");
Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 2.9.



Рисунок 2.9 – Результати розв’язання задачі 2.6

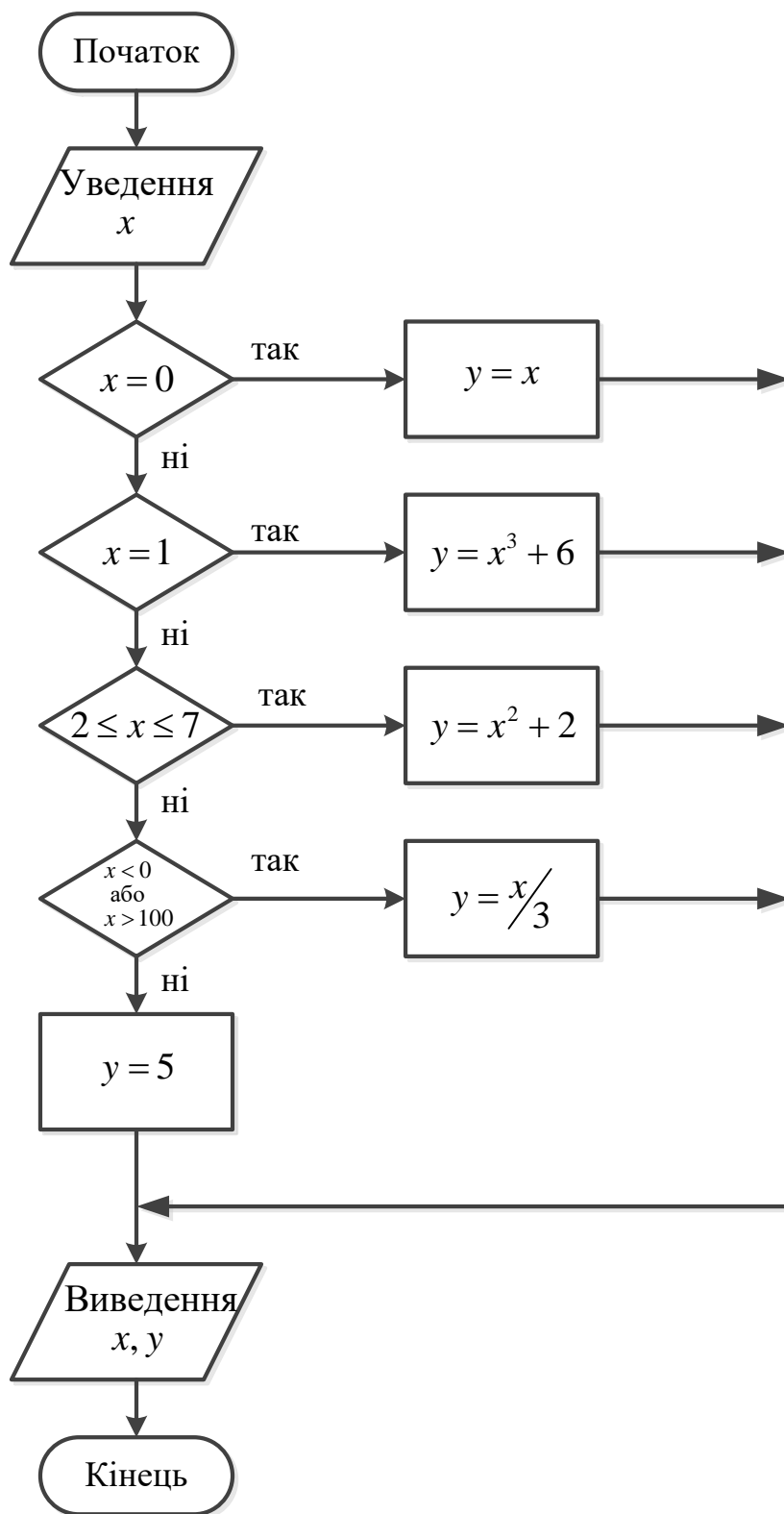


Рисунок 2.10 – Блок-схема алгоритму до задачі 2.6

Задача 2.7. Приклад програми, що запитує у користувача номер дня тижня і потім виводить його назву. Якщо уведено неправильні дані, програма повинна вивести повідомлення про помилку. Використовується оператор `switch`.

Програмний код:

```
using System;
namespace C8
{
    class Program
    {
        static void Main(string[] args)
        {
            int nom;
            Console.Write(" Введіть номер дня тижня ? ");
            nom = Convert.ToInt32(Console.ReadLine());
            switch (nom)
            {
                case 1:
                    Console.WriteLine(" Понеділок");
                    break;

                case 2:
                    Console.WriteLine(" Вівторок");
                    break;

                case 3:
                    Console.WriteLine(" Середа");
                    break;

                case 4:
                    Console.WriteLine(" Четвер");
                    break;

                case 5:
                    Console.WriteLine(" П'ятниця");
                    break;

                case 6:
                    Console.WriteLine(" Субота");
                    break;

                case 7:
                    Console.WriteLine(" Неділя");
                    break;

                default:
                    Console.WriteLine(" Число має бути в діапазоні 1..7");
                    break;
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 2.11.

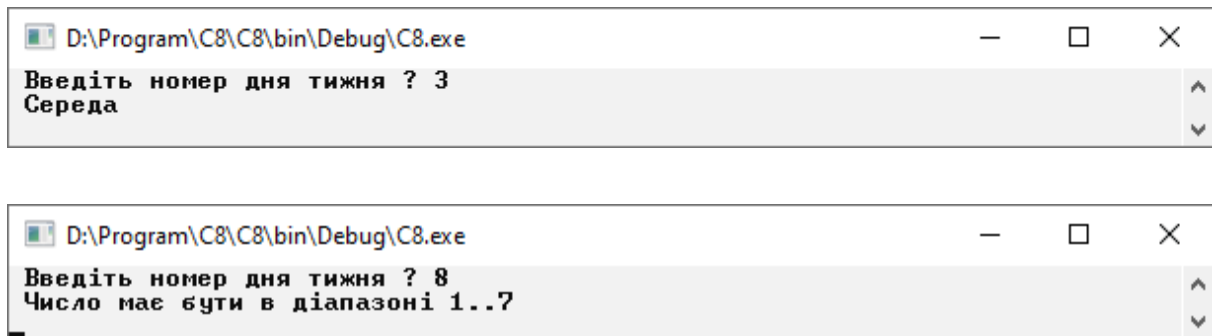


Рисунок 2.11 – Результати розв’язання задачі 2.7

Задача 2.8. Програма запитує у користувача номер місяця і потім виводить відповідну назву пори року. Якщо користувач уводить неприпустиме число, програма виводить повідомлення «Число має бути в діапазоні 1..12».

Програмний код:

```
using System;

namespace C8a
{
    class Program
    {
        static void Main(string[] args)
        {
            int M;
            Console.Write(" Введіть номер поточного місяця ? ");
            M = Convert.ToInt32(Console.ReadLine());
            switch (M)
            {
                case 12:
                case 1:
                case 2:
                    Console.WriteLine(" Зима");
                    break;

                case 3:
                case 4:
                case 5:
                    Console.WriteLine(" Весна");
                    break;

                case 6:
                case 7:
                case 8:
```

```

        Console.WriteLine(" Літо");
        break;

    case 9:
    case 10:
    case 11:
        Console.WriteLine(" Осінь");
        break;

    default:
        Console.WriteLine(" Число має бути в діапазоні 1..12");
        break;
    }
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 2.12.

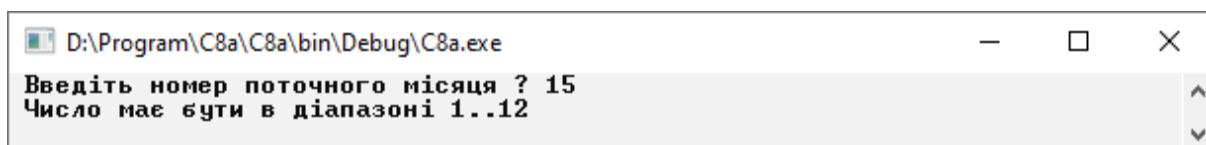
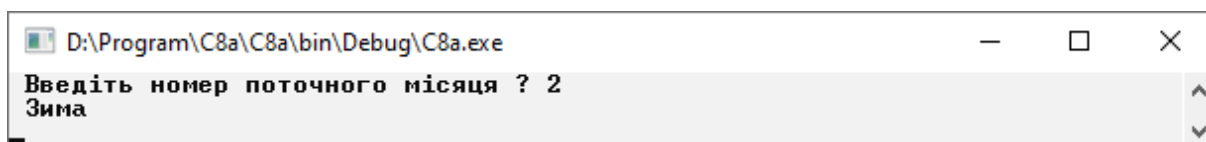


Рисунок 2.12 – Результати розв’язання задачі 2.8

Задача 2.9. У цьому завданні вводиться вік людини і в консольне вікно виводиться вікова група, до якої належить людина. Застосовуються оператори `if` і `switch`.

Програмний код:

```

using System;

namespace C133
{
    class Program
    {
        static void Main(string[] args)
        {
            int age; // вік людини
            Console.WriteLine(" Введіть вік ? ");
            age = Convert.ToInt32(Console.ReadLine());

```

```

int p = 0;
if ((age >= 0) && (age < 7)) p = 1;
if ((age >= 7) && (age < 17)) p = 2;
if ((age >= 17) && (age < 25)) p = 3;
if ((age >= 25) && (age < 50)) p = 4;
if ((age >= 50) && (age < 80)) p = 5;
if (age >= 80) p = 6;
switch (p)
{
    case 1:
        Console.WriteLine(" дитина");
        break;
    case 2:
        Console.WriteLine(" школяр");
        break;
    case 3:
        Console.WriteLine(" юнак");
        break;
    case 4:
        Console.WriteLine(" доросла людина");
        break;
    case 5:
        Console.WriteLine(" літня людина");
        break;
    case 6:
        Console.WriteLine(" стара людина");
        break;
    default:
        Console.WriteLine(" Ви ввели від'ємне число ");
        break;
}
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 2.13.

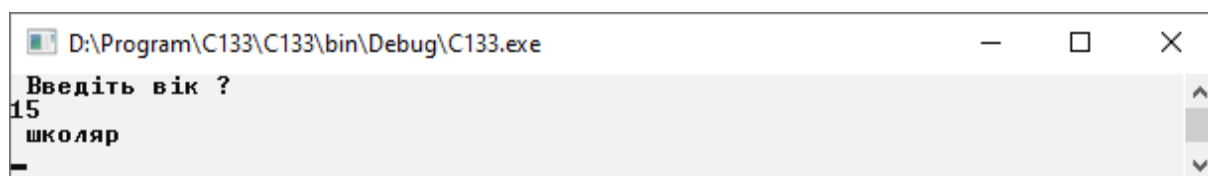


Рисунок 2.13 – Результати розв'язання задачі 2.9

2.2 Завдання для виконання лабораторних робіт (частина 1)

Обчислити значення функції. Варіанти задач наведено у табл. 2.1.

Таблиця 2.1 – Варіанти задач

Вар.	Функція	Початкові дані
1	2	3
1	$z = \begin{cases} \sqrt{ x^2 + 2x + y }, & x < 0, y < 0 \\ x + y, & \text{якщо } x > 0, y > 0 \\ x - y , & \text{в інших випадках} \end{cases}$	$x = -0,5; y = -2,5$ $x = 2,31; y = 4,2$ $x = 5; y = -7$
2	$y = \begin{cases} \sin^2 x, & x > 0 \\ \ln^3(1 + x^2), & \text{якщо } x < 0 \\ x + c, & x = 0 \end{cases}$	$c = 1,57$ $x = 3; -2,5; 0$
3	$g = \begin{cases} y^3 - 0,3, & y < 0 \\ 0, & \text{якщо } 0 \leq y \leq 1 \\ y^2 + y, & y > 1 \end{cases}$	$y = z + 2;$ $z = -3; -1,5; 0$
4	$c = \begin{cases} (xy)^z & y > 0, z > 0 \\ (x + 5y + z)^2 & \text{якщо } y < 0, z < 0 \\ 0 & \text{решта} \end{cases}$	$y = 1; -0,5; 0$ $z = 0,5; -1,5; 3$ $x = 0,5$
5	$m = \begin{cases} (t^2 + 1)bc, & t < 0 \\ 1/(t + 1), & \text{якщо } 0 \leq t \leq 4 \\ \sin(t + 1)^2, & t \geq 4 \end{cases}$	$b = 0,5; c = 1,2;$ $t = -2; 3; 5,67$
6	$y = \begin{cases} at^2 \ln t, & 1 \leq t \leq 2 \\ 1, & \text{якщо } t < 1 \\ e^{at} \cos bt, & t > 2 \end{cases}$	$a = -0,5$ $b = 2$ $t = 1,5; 0,5; 2,3$
7	$\omega = \begin{cases} x\sqrt[3]{x - a}, & x > a \\ x \sin ax, & \text{якщо } x = a \\ e^{-ax} \cos ax, & x < a \end{cases}$	$a = 2,5$ $x = 3; 2,5; 1$

Продовження табл. 2.1

1	2	3
8	$z = \begin{cases} 4a^2 + b, & a < 0, b < 0 \\ 2 \cos a - \ln b, & \text{якщо } a > 0, b > 0 \\ a + 12, & a < 0, b > 0 \end{cases}$	$a = -2; \quad b = -5;$ $a = 3,5; \quad b = 7;$ $a = -8; \quad b = 12;$
9	$y = \begin{cases} x, & x < 1 \\ x^3 + 6, & \text{якщо } 1 \leq x < 2 \\ x^2 + 2, & x = 2 \\ 0, & x > 2 \end{cases}$	$x = 0,5; 1,2; 2; 4$
10	$g = \begin{cases} ax^3 + a \cos^2 x , & x > 0 \\ ax^2 + bx, & \text{якщо } x = 0 \\ b, & x < 0 \text{ і } x > 2 \\ ax + b, & \text{в інших випадках} \end{cases}$	$a = 2; \quad b = 1$ $x = 2; 0; -3,5; -1,5$

2.3 Завдання для виконання лабораторних робіт (частина 2)

Завдання 1. Написати програму, яка запитує у користувача номер місяця і потім виводить відповідну назву пори року. Якщо користувач введе неприпустиме число, програма повинна вивести повідомлення «Помилка даних». Програму написати двома засобами:

- з використанням вкладених операторів `if`;
- з використанням оператора вибору `switch`.

Нижче наведено рекомендований вид екрану програми

Введіть номер місяця (число від 1 до 12)

? 12

Зима

Завдання 2. Написати програму, яка дозволяє порахувати вартість заправки автомобіля.

Початкові дані:

- тип палива (бензин 92, 95, 98 або дизельне паливо);
- кількість літрів.

Використовувати оператор вибору `switch`.

Нижче наведено рекомендований вид екрану програми (дані, введені користувачем, виділені напівжирним шрифтом).

Бензин

Марка бензину:

1 – 92

2 – 95

3 – 98
4 – DT
Ваш вибір ? 2
Літрів? 25

Ціна за 1 літр: 10 грн.
Літрів: 25

До оплати: 250 грн.

Завдання 3. Написати програму, яка дозволяє порахувати вартість друку фотографій.

Початкові дані:

- розмір фотографій (9×12 , 10×15 або 18×24);
- кількість фотографій.

Якщо замовлених фотографій більше 10, замовнику надається знижка – 10%, якщо замовлених фотографій більше 50, знижка –15%.

Використовувати оператор `if`.

Нижче наведено рекомендований вид екрану програми (дані, введені користувачем, виділені напівжирним шрифтом).

Фото

Розмір:

1 – 9×12

2 – 10×15

3 – 18×24

Ваш вибір ? 1

Кількість? 12

Ціна: 3,50 грн.

Кількість: 12 шт.

Сума: 42,00 грн.

Знижка: 4,20 грн.

До сплати: 37,80 грн.

Завдання 4. Написати програму, яка запитує у користувача номер дня тижня і виводить одне з повідомлень: "Робочий день", "Субота" або "Неділя". Використовувати вкладені `if`.

Завдання 5. Написати програму обчислення вартості розмови по телефону з урахуванням 20% -ної знижки, що надається по суботах і неділях. Нижче наведено рекомендований вид екрану програми:

Обчислення вартості розмови по телефону

Введіть початкові дані:

Тривалість розмови (ціла кількість хвилин) -> 3

День тижня (1 – понеділок, 7 – неділя) -> 6

Надається знижка 20%

Вартість розмови: 15,52 грн.

Тема 3. ЦИКЛИ

Мета роботи: вивчити оператори циклу: for, while, do-while.

3.1 Приклади розв'язання задач

Задача 3.1. Вивести на консоль значення перших десяти цілих чисел за допомогою оператора циклу for.

Програмний код:

```
using System;

namespace C13
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            for (i = 1; i <= 10; i++) Console.WriteLine(" " + i);
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.1.

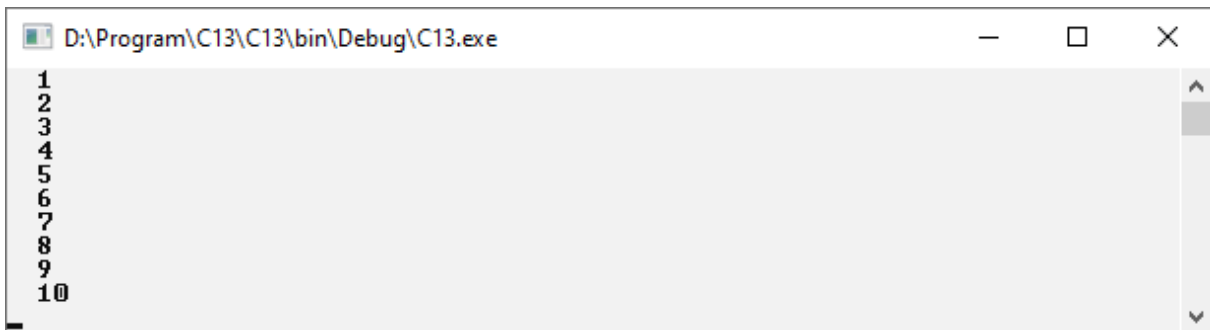


Рисунок 3.1 – Результати розв'язання задачі 3.1

Задача 3.2. Програма виводить числа, поступово дедалі менші від 20 до -20 на величину 5.

Програмний код:

```
using System;

namespace C13a
{
```

```

class Program
{
    static void Main(string[] args)
    {
        int k;
        for (k = 20; k >= -20; k = k - 5)
            Console.WriteLine(" " + k);
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 3.2.

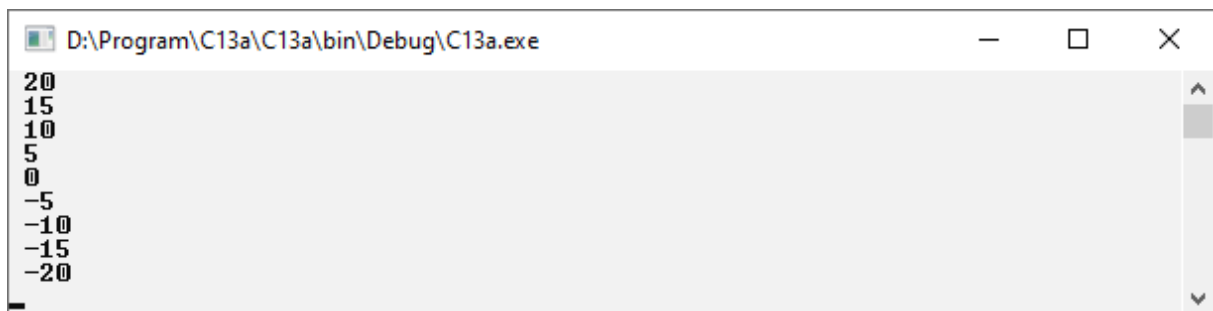


Рисунок 3.2 – Результати розв’язання задачі 3.2

Задача 3.3. Використання в операторі `for` двох або більше змінних для керування циклом

Програмний код:

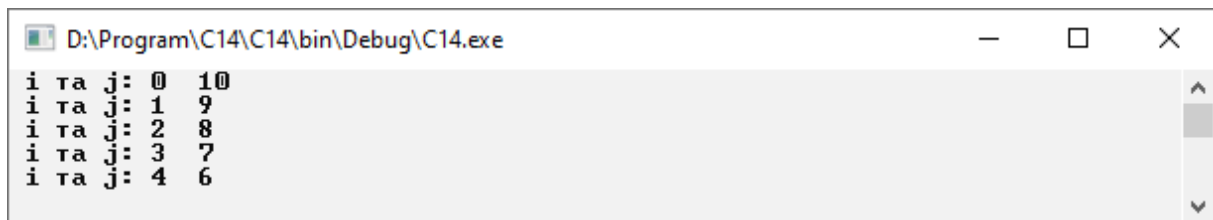
```

using System;

namespace C14
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, j;
            for (i = 0, j = 10; i < j; i++, j--)
                Console.WriteLine(" i та j: " + i + " " + j);
            Console.ReadLine();
        }
    }
}

```

Результати розрахунку подано на рис. 3.3.



```
D:\Program\C14\C14\bin\Debug\C14.exe
i та j: 0 10
i та j: 1 9
i та j: 2 8
i та j: 3 7
i та j: 4 6
```

Рисунок 3.3 – Результати розв’язання задачі 3.3

Примітка. Коли цикл починається, ініціалізуються обидві змінні *i* та *j*. Щоразу, коли цикл повторюється, змінна *i* інкрементується, а змінна *j* декрементується.

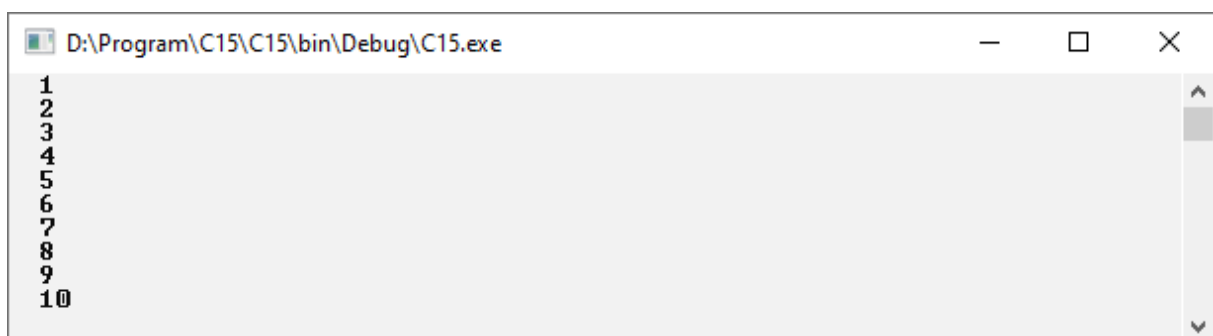
Задача 3.4. Вивести на консоль значення перших десяти цілих чисел за допомогою оператора циклу `while`.

Програмний код:

```
using System;

namespace C15
{
    class Program
    {
        static void Main(string[] args)
        {
            int i = 1;
            while (i <= 10)
            {
                Console.WriteLine(" " + i);
                i++;
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.4.



```
D:\Program\C15\C15\bin\Debug\C15.exe
1
2
3
4
5
6
7
8
9
10
```

Рисунок 3.4 – Результати розв’язання задачі 3.4

Пояснення. У цій задачі до початку циклу `while` змінній *i* присвоюється значення 1. У циклі спочатку перевіряється значення *i*, якщо воно менше або дорівнює 10, то змінна *i* інкрементується та значення *i* виводиться на екран. Цикл повторюється доти, доки значення змінної *i* менше або дорівнює 10. Як тільки воно виявиться більше 10, цикл завершується.

Задача 3.5. Вивести на консоль значення перших десяти цілих чисел за допомогою оператора циклу do-while.

Програмний код:

```
using System;

namespace C17
{
    class Program
    {
        static void Main(string[] args)
        {
            int i = 1;
            do
            {
                Console.Write(" " + i);
                i++;
            }
            while (i <= 10);
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.5.



Рисунок 3.5 – Результати розв’язання задачі 3.5

Задача 3.6. Використання оператора break для виходу з циклу

Програмний код:

```
using System;

namespace C18
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = -5; i <= 5; i++)
            {
                if (i > 0) break; // завершити цикл, як тільки
                                // змінна i стане додатною
                Console.Write(" " + i);
            }
        }
    }
}
```

```

        Console.WriteLine();
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 3.6.



Рисунок 3.6 – Результати розв’язання задачі 3.6

Примітка. В цьому прикладі цикл `for` організований для виконання в межах від `-5` до `5`, але незважаючи на це, оператор `break` перериває його раніше, коли значення змінної `i` стає додатним.

Задача 3.7. Оператор `continue` використовується як допоміжний засіб для виведення парних чисел у межах від `0` до `10`.

Програмний код:

```

using System;

namespace C19
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 0; i <= 10; i++)
            {
                if ((i % 2) != 0) continue; //перейти до наст. кроку
                // ітерації

                Console.Write(" " + i);
            }
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}

```

Результати розрахунку подано на рис. 3.7.



Рисунок 3.7 – Результати розв’язання задачі 3.7

Задача 3.8. Написати програму, яка виводить таблицю значень функції $y = -2,4x^2 + 5x - 3$ в діапазоні від -2 до 2 , з кроком $0,5$. Таблиця повинна складатися з двох стовпців: значень аргументу і відповідних йому значень функції. Для розв’язання задачі використовувати оператори циклів: `for`, `while`, `do-while`.

Спосіб 1. Створити консольний додаток із використанням циклу `for`

Програмний код:

```
using System;

namespace C9
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y;           // аргумент і значення функції
            double xn = -2.0;      // початкове значення аргументу x
            double xk = 2.0;       // кінцеве значення аргументу x
            double dx = 0.5;       // крок змінення аргументу
            Console.WriteLine("-----");
            Console.WriteLine("  x          y  ");
            Console.WriteLine("-----");
            for (x = xn; x <= xk; x = x + dx)
            {
                y = -2.4 * x * x + 5 * x - 3;
                Console.WriteLine($" {x,7:F2}  {y,9:F3}");
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.8.

x	y
-2,00	-22,600
-1,50	-15,900
-1,00	-10,400
-0,50	-6,100
0,00	-3,000
0,50	-1,100
1,00	-0,400
1,50	-0,900
2,00	-2,600

Рисунок 3.8 – Результати розв’язання задачі 3.8 (Спосіб 1)

Пояснення:

1) `Console.WriteLine($" {x,7:F2} {y,9:F3}");`

Виводить рядок результатів у заданому форматі в консольне вікно.

Примітка: Щоб зупинити програму, яка зациклила, використовуйте клавіші `<Esc>` або `<Ctrl> + <Break>`.

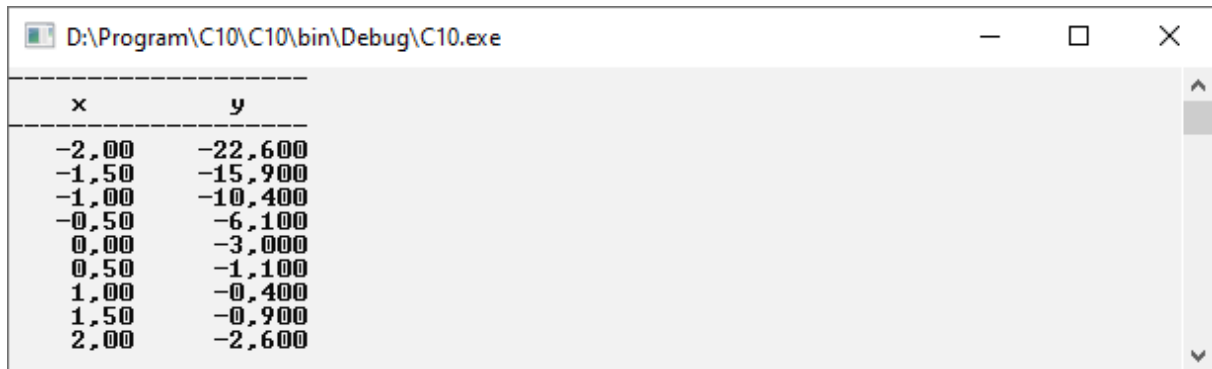
Спосіб 2. Створити консольний додаток із використанням циклу `while`.

Програмний код:

```
using System;

namespace C10
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y;           // аргумент і значення функції
            double xn = -2.0;      // початкове значення аргументу x
            double xk = 2.0;       // кінцеве значення аргументу x
            double dx = 0.5;       // крок змінення аргументу
            Console.WriteLine("-----");
            Console.WriteLine("  x          y  ");
            Console.WriteLine("-----");
            x = xn;
            while (x <= xk)
            {
                y = -2.4 * x * x + 5.0 * x - 3.0;
                Console.WriteLine($" {x,7:F2} {y,9:F3}");
                x = x + dx;
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.9.



x	y
-2,00	-22,600
-1,50	-15,900
-1,00	-10,400
-0,50	-6,100
0,00	-3,000
0,50	-1,100
1,00	-0,400
1,50	-0,900
2,00	-2,600

Рисунок 3.9 – Результати розв’язання задачі 3.8 (Спосіб 2)


Спосіб 3. Створити консольний додаток із використанням циклу do-while.

Програмний код:

```
using System;

namespace C11
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y;          // аргумент і значення функції
            double xn = -2;      // початкове значення аргументу x
            double xk = 2;       // кінцеве значення аргументу x
            double dx = 0.5;     // крок змінення аргументу
            Console.WriteLine("-----");
            Console.WriteLine("    x          y    ");
            Console.WriteLine("-----");
            x = xn;
            do
            {
                y = -2.4 * x * x + 5 * x - 3;
                Console.WriteLine($" {x,7:F2}  {y,9:F3}");
                x = x + dx;
            }
            while (x <= xk);
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.10.



x	y
-2,00	-22,600
-1,50	-15,900
-1,00	-10,400
-0,50	-6,100
0,00	-3,000
0,50	-1,100
1,00	-0,400
1,50	-0,900
2,00	-2,600

Рисунок 3.10 – Результати розв’язання задачі 3.8 (Спосіб 3)

Задача 3.9. Створити консольний додаток для обчислення факторіала $k!$

Програмний код:

```
using System;
namespace C12
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, k;
            double p;
            Console.Write(" Уведіть k= ? ");
            k = Convert.ToInt32(Console.ReadLine());
            p = 1;
            for (i = 1; i <= k; i++)
                p = p * i;
            Console.WriteLine($" {k}! = {p}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.11.



Уведіть k= ? 9
9! = 362880

Рисунок 3.11 – Результати розв’язання задачі 3.9

Задача 3.10. Написати програму, яка обчислює суму перших n членів ряду: $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$

Кількість членів ряду, що додаються, задають під час виконання програми.

Програмний код:

```
using System;

namespace C16a
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;          // кількість членів ряду, що додаються
            int i;          // номер елемента ряду
            double elem;    // значення елемента ряду
            double sum;     // сума елементів ряду
            Console.Write(" Введіть кількість членів ряду, що додаються n=? ");
            n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine(" n= " + n);
            sum = 0;
            for (i = 1; i <= n; i++)
            {
                elem = 1.0 / i;
                sum = sum + elem;
            }
            Console.WriteLine();
            Console.WriteLine($" Сума перших {n} членів ряду =
                                {sum:F4}");

            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.12

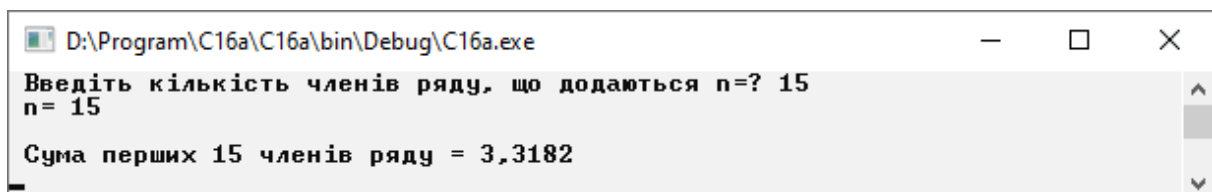


Рисунок 3.12 – Результати розв’язання задачі 3.10

Задача 3.11. Вивести для аргументу x , який змінюється в межах $-2 \leq x \leq 12$ із кроком 2, таблицю значень такої функції:

$$y = \begin{cases} t, & x < 0 \\ t \cdot x, \text{ якщо } & 0 \leq x \leq 10 \\ 2t, & x \geq 10 \end{cases}$$

за умови $t = 5$.

Програма повинна виводити таблицю, яка складається з двох стовпців: значень аргументу і значень функції, що їм відповідає.

Блок-схему розв'язанні задачі наведено на рис. 3.13.

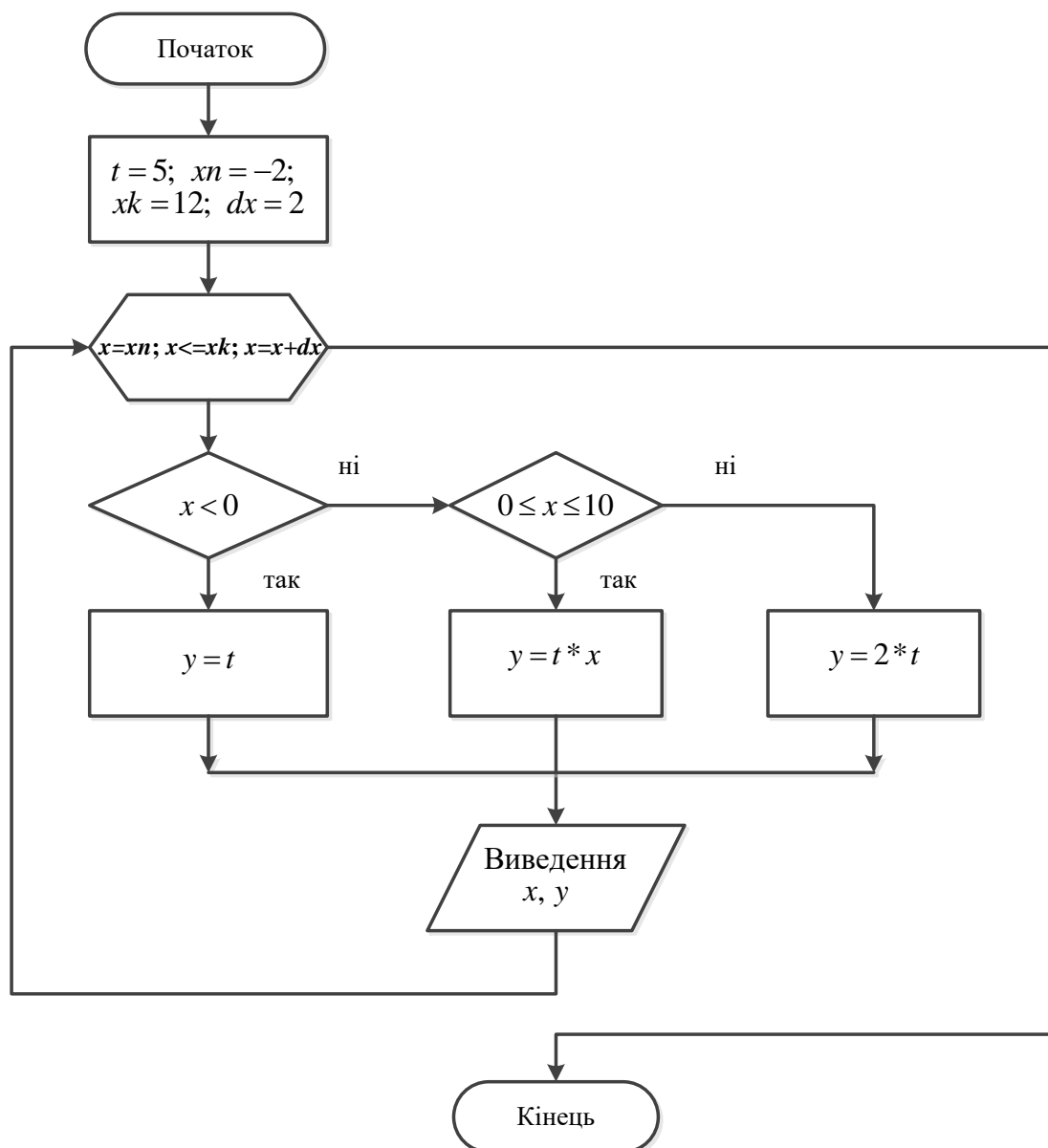


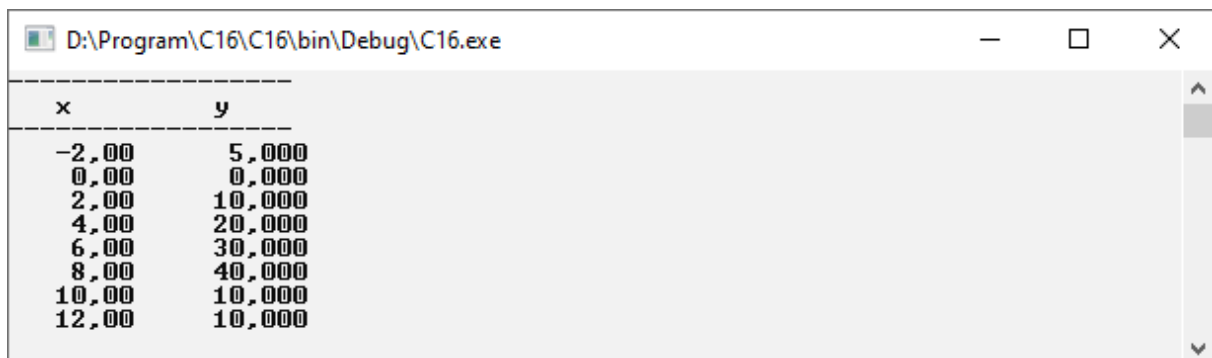
Рисунок 3.13 – Блок-схема алгоритму до задачі 3.11

Програмний код:

```
using System;

namespace C16
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y, t = 5;
            double xn = -2;    // початкове значення аргументу
            double xk = 12;   // кінцеве значення аргументу
            double dx = 2;    // крок змінення аргументу
            // заголовок таблиці
            Console.WriteLine("-----");
            Console.WriteLine("  x          y  ");
            Console.WriteLine("-----");
            // обчислення та виведення таблиці
            for (x = xn; x <= xk; x = x + dx)
            {
                if (x < 0)
                    y = t;
                else if (0 <= x && x < 10)
                    y = t * x;
                else
                    y = 2 * t;
                Console.WriteLine($" {x,7:F2}  {y,9:F3}");
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 3.14.



x	y
-2,00	5,000
0,00	0,000
2,00	10,000
4,00	20,000
6,00	30,000
8,00	40,000
10,00	10,000
12,00	10,000

Рисунок 3.14 – Результати розв’язання задачі 3.11

3.2 Завдання для виконання лабораторних робіт (частина 1)

Задача 1. Написати програму, яка виводить таблицю значень функції:

$$y = x + bx \operatorname{Sin} L \quad \text{для } 0 \leq x \leq 1, \text{ з кроком } h = 0,1;$$

де $b = 2,3; L = 1,45$.

Задача 2. Написати програму, яка виводить таблицю значень функції:

$$y = x + 3x^2 - 2,5x^3 + 0,75x^4 \quad \text{для } 1 \leq x \leq 4, \text{ з кроком } h = 0,5.$$

Задача 3. Написати програму, яка виводить таблицю значень функції:

$$y = 8m + 4m^4 - 1,5m^3 \quad \text{для } 1 \leq m \leq 3, \text{ з кроком } h = 0,2.$$

Задача 4. Написати програму, яка виводить таблицю значень функції:

$$S1 = 4\pi r^2; \quad S2 = \pi r^2 / 2 \quad \text{для } 1 \leq r \leq 1,4; \text{ з кроком } h = 0,05.$$

Результат отримати у вигляді таблиці

R	S1	S2
1.00	12.566	1.571
.	.	.
.	.	.
.	.	.
.	.	.
1.40	24.630	3.079

Задача 5. Написати програму, яка виводить таблицю значень функції:

$$y = |x| \quad \text{для } -4 \leq x \leq 4, \text{ з кроком } h = 0,5.$$

Задача 6. Написати програму, яка виводить таблицю значень функції:

$$y = |x - 2| + |x + 1| \quad \text{для } -4 \leq x \leq 4, \quad \text{з кроком } h = 0,5.$$

Задача 7. Написати програму, яка виводить таблицю значень функції:

$$y = 2x^3 - 15x^2 + 4,7x - 3 \quad \text{для } -3 \leq x \leq 3, \quad \text{з кроком } h = 0,5.$$

Задача 8. Написати програму, яка виводить таблицю значень функції:

$$y = 5^{\cos(x)} + 5 \sin(x)^2 - 3 \quad \text{для } 1 \leq x \leq 5, \quad \text{з кроком } h = 0,5.$$

Задача 9. Написати програму, яка виводить таблицю значень функції:

$$y = \frac{3 \sin(x)^3 + 2,3}{x - 2 \cos(x)^3} \quad \text{для } 3 \leq x \leq 9, \quad \text{з кроком } h = 0,5.$$

Задача 10. Написати програму, яка виводить таблицю значень функції:

$$y = (\sin(x)^3 - 0,6) / x^2 \quad \text{для } 1 \leq x \leq 10, \quad \text{з кроком } h = 1.$$

3.3 Завдання для виконання лабораторних робіт (частина 2)

Задача 1. Сума послідовності.

Написати програму: з клавіатури увести ціле додатне число n . На консоль вивести суму всіх цілих чисел від 0 до n .

Задача 2. Добуток послідовності.

Написати програму: з клавіатури увести два цілих додатних числа $n1$ і $n2$. На консоль вивести добуток всіх цілих чисел від $n1$ до $n2$ включно.

Задача 3. Написати програму, яка виводить таблицю результатів піднесення чисел до квадрата та куба. Обчислити для перших десяти цілих додатних чисел. Нижче наведено рекомендований вигляд екрана програми.

Число	Квадрат	Куб
1	1	1
2	4	8
3	9	27
·	·	·
·	·	·
10	100	1000

Задача 4. Сума послідовності.

Написати програму: з клавіатури увести дійсне додатне число x з точністю до 0,1. На консоль вивести суму всіх чисел від 0 до x з кроком 0,1. Задачу розв'язати двома засобами – з використанням операторів `for` і `while`.

Задача 5. Написати програму, яка виводить таблицю швидкості (через кожні 0,5 с) тіла, що вільно спадає $v = g \cdot t$, де $g = 9,8 \text{ м/с}^2$ – прискорення вільного падіння.

Рекомендований вигляд екрана наведено нижче.

Час, с	Швидкість, м/с
0.0	0.00
0.5	4.90
1.0	9.80
1.5	14.70
2.0	19.60
2.5	24.50
3.0	29.40

Задача 6. Написати програму, яка виводить на екран таблицю відповідності температури у градусах Цельсія та Фаренгейта. ($F = (9,0/5,0) \cdot C + 32$). Діапазон змінення температури у градусах Цельсія і крок повинні бути уведені під час роботи програми.

Рекомендований вигляд екрана наведено нижче.

t1 =? 0
t2 =? 10
dt =? 1

C	F

0.0	32.00
1.00	33.80
2.00	35.60
.	.
.	.
10.00	50.00

Задача 7. Написати програму, яка виводить на екран таблицю множення, наприклад, на 7. Рекомендований вигляд екрана наведено нижче.

7×1=7
7×2=14
7×3=21
7×4=28
7×5=35
7×6=42
7×7=49
7×7=56
7×9=63
7×10=70

Тема 4. Одновимірні масиви

Мета роботи: вивчити оператори оголошення масивів, способи наповнення масивів, введення-виведення елементів масивів, накопичення суми і добутку елементів масивів, сортування елементів масиву, а також знаходження максимального елемента одновимірного масиву і його індексу.

4.1 Приклади розв'язання задач

Задача 4.1. Виконати введення та виведення елементів одновимірного масиву $Y(12)$.

Пояснення: Позначка «одновимірний масив $Y(12)$ » означає масив $Y = (y_i; i = 0, 1, \dots, 11)$, розмір масиву (кількість елементів) = 12. Нижній індекс елементів масиву = 0, верхній індекс = 11.

Спосіб 1. Створимо консольний додаток, в якому елементи масиву задаються в програмному коді. Розмір масиву зазначимо за допомогою константи.

Програмний код:

```
using System;

namespace C21
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву
            int i;
            double[] Y = new double[n] { 1.0, 2.5, 3.7, 4.8, 5.0,
                                         6.2, 7.3, 8.0, 9.5, 10.0, 12.0, 14.0 };
            Console.WriteLine(" Елементи масиву: ");
            // Виведення елементів масиву
            for (i = 0; i < n; i++)
                Console.WriteLine(" " + Y[i]);
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.1.

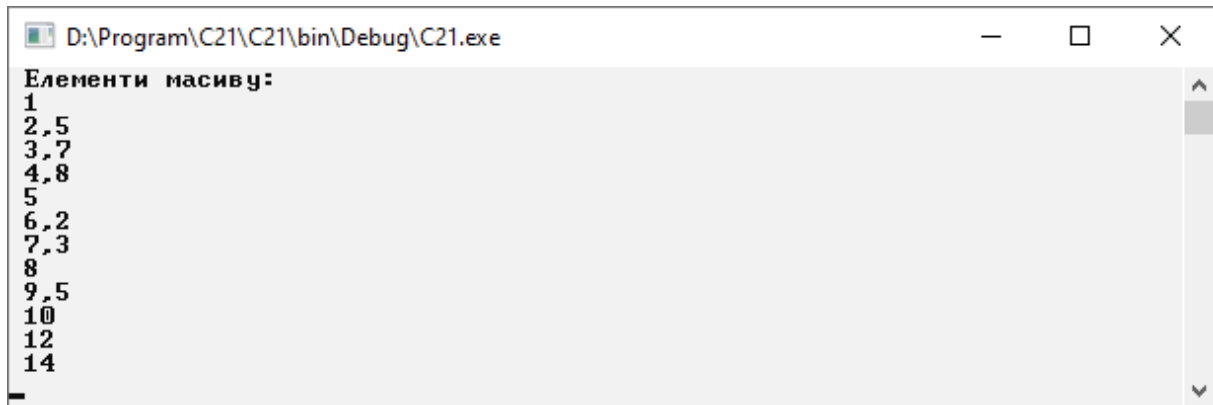


Рисунок 4.1 – Результати розв’язання задачі 4.1 (спосіб 1)

Пояснення:

```
1)double[] Y = new double[n] { 1.0, 2.5, 3.7, 4.8, 5.0, 6.2,  
                               7.3, 8.0, 9.5, 10.0, 12.0, 14.0 };
```

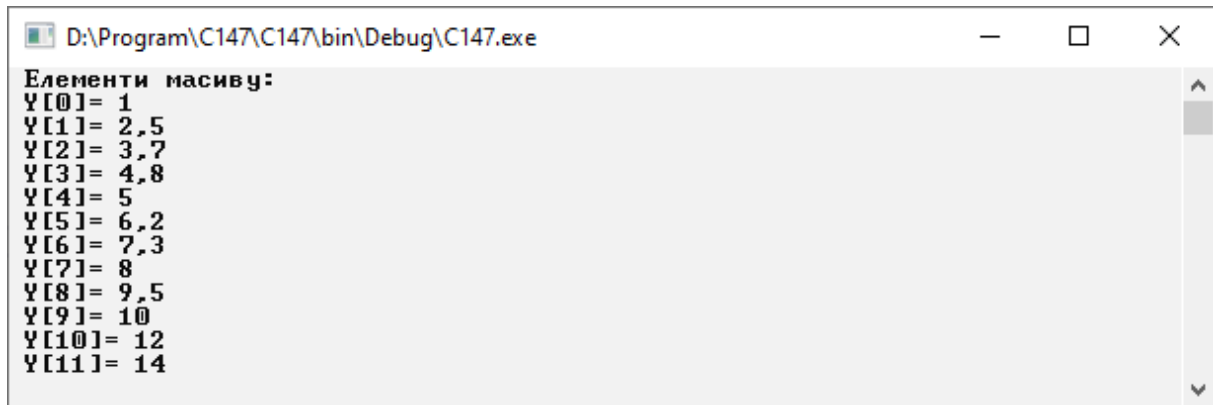
Одночасно із створенням масиву він заповнюється потрібними значеннями. Якщо зазначено і розмір n, і перелік ініціалізаторів, то розмір n повинен бути тільки константою.

Спосіб 1А. Створимо консольний додаток, в якому елементи масиву задаються в програмному коді. Розмір масиву зазначимо за допомогою методу Length (довжина масиву – кількість елементів масиву).

Програмний код:

```
using System;  
  
namespace C147  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            int i;  
            double[] Y = { 1.0, 2.5, 3.7, 4.8, 5.0, 6.2, 7.3, 8.0,  
                          9.5, 10.0, 12.0, 14 };  
            Console.WriteLine(" Елементи масиву: ");  
            for (i = 0; i < Y.Length; i++)  
                Console.WriteLine($" Y[{i}] = {Y[i]}");  
            Console.ReadLine();  
        }  
    }  
}
```

Результати розрахунку подано на рис. 4.2.



```
D:\Program\C147\C147\bin\Debug\C147.exe
Елементи масиву:
Y[0]= 1
Y[1]= 2,5
Y[2]= 3,7
Y[3]= 4,8
Y[4]= 5
Y[5]= 6,2
Y[6]= 7,3
Y[7]= 8
Y[8]= 9,5
Y[9]= 10
Y[10]= 12
Y[11]= 14
```

Рисунок 4.2 – Результати розв’язання задачі 4.1 (спосіб 1А)

Спосіб 2. Створимо консольний додаток, у якому елементи масиву розраховуються за формулою:

$$y_i = i/2.0 + 7$$

Програмний код:

```
using System;

namespace C23
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву
            double[] Y = new double[n];
            int i;
            Console.WriteLine(" Елементи масиву: ");
            for (i = 0; i < n; i++)
            {
                Y[i] = i / 2.0 + 7;
                Console.WriteLine($" Y[{i}]={Y[i]}");
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.3.

```
D:\Program\C23\C23\bin\Debug\C23.exe
Елементи масиву:
Y[0]=7
Y[1]=7,5
Y[2]=8
Y[3]=8,5
Y[4]=9
Y[5]=9,5
Y[6]=10
Y[7]=10,5
Y[8]=11
Y[9]=11,5
Y[10]=12
Y[11]=12,5
```

Рисунок 4.3 – Результати розв’язання задачі 4.1 (спосіб 2)

Пояснення:

1) `double [] Y = new double[n];`

Оголошується масив `Y` із `n` дійсних чисел типу `double`, при цьому всі `n` елементів автоматично (неявно) ініціалізуються нулями;

`n` може бути не тільки константою, але й виразом типу, що приводить до цілого.

2) `Y[i] = i/2.0 + 7;`

Якщо написати `Y[i] = i/2+7;` отримаємо ділення цілих чисел. При діленні цілих чисел остача буде відкинута, щоб цього не відбулося необхідно змінювати тип хоча б одного з цих чисел, тобто `i/2.0`.

Спосіб 3. Створимо консольний додаток, у якому елементи масиву вводяться з клавіатури.

На рис. 4.5, 4.6 зображено блок-схеми уведення-виведення елементів одновимірного масиву.

Програмний код:

```
using System;

namespace C24
{
    class Program
    {
        static void Main(string[] args)
        {
            int n; //кількість елементів масиву
            int i;
            Console.Write(" Введіть n=? ");
            n = Convert.ToInt32(Console.ReadLine());
            double[] Y = new double[n];
            Console.WriteLine("\n Введення елементів масиву: ");
```

```

for (i = 0; i < n; i++)
{
    Console.WriteLine($" Введіть {i} елемент масиву =? ");
    Y[i] = Convert.ToDouble(Console.ReadLine());
}
Console.WriteLine();
Console.WriteLine(" Виведення елементів масиву: ");
for (i = 0; i < n; i++)
{
    Console.WriteLine($" Y[{i}]={Y[i]}");
}
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 4.4

```

D:\Program\C24\C24\bin\Debug\C24.exe
Введіть n=? 12
Введення елементів масиву:
Введіть 0 елемент масиву =? 1,0
Введіть 1 елемент масиву =? 2,5
Введіть 2 елемент масиву =? 3,7
Введіть 3 елемент масиву =? 4,8
Введіть 4 елемент масиву =? 5,0
Введіть 5 елемент масиву =? 6,2
Введіть 6 елемент масиву =? 7,3
Введіть 7 елемент масиву =? 8,0
Введіть 8 елемент масиву =? 9,5
Введіть 9 елемент масиву =? 10,0
Введіть 10 елемент масиву =? 12,0
Введіть 11 елемент масиву =? 14,0
Виведення елементів масиву:
Y[0]=1
Y[1]=2,5
Y[2]=3,7
Y[3]=4,8
Y[4]=5
Y[5]=6,2
Y[6]=7,3
Y[7]=8
Y[8]=9,5
Y[9]=10
Y[10]=12
Y[11]=14

```

Рисунок 4.4 – Результати розв’язання задачі 4.1 (спосіб 3)

Пояснення:

```
Console.WriteLine("\nВведення елементів масиву: ");
```

Рядкова константа "\nВведення елементів масиву: " містить керуючу послідовність символів \n – переведення рядка.

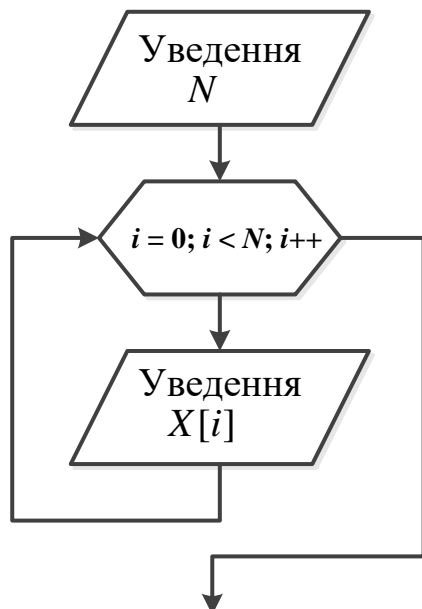


Рисунок 4.5 – Блок-схема уведення елементів масиву

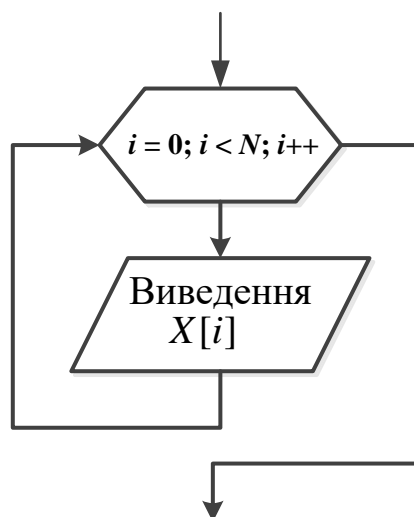


Рисунок 4.6 – Блок-схема виведення елементів масиву

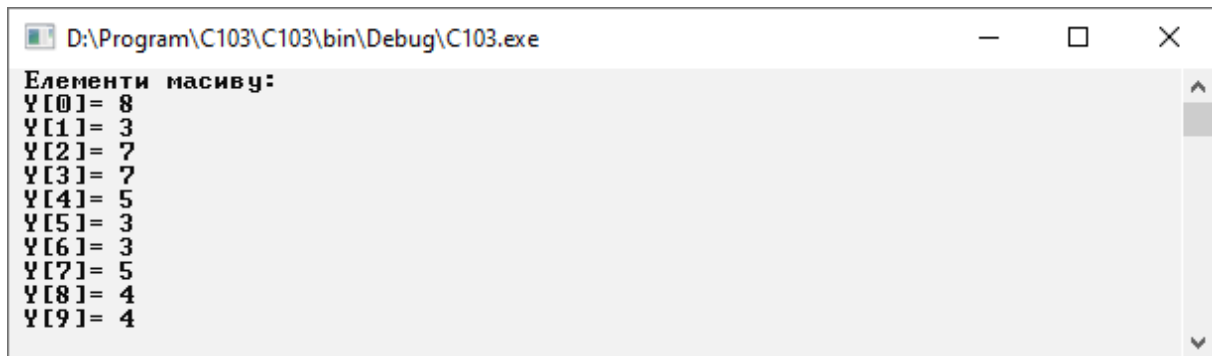
Спосіб 4А. Створимо консольний додаток, у якому елементи масиву цілого типу задаються за допомогою генератора випадкових чисел.

Програмний код:

```
using System;

namespace C103
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 10; // кількість елементів масиву
            int[] Y = new int[n];
            int i;
            //створюється екземпляр rnd класу Random
            Random rnd = new Random();
            Console.WriteLine(" Елементи масиву: ");
            for (i = 0; i < n; i++)
            {
                Y[i] = rnd.Next(1, 10);
                Console.WriteLine($" Y[{i}] = {Y[i]:D}");
            }
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.7.



```
D:\Program\C103\C103\bin\Debug\C103.exe
Елементи масиву:
Y[0]= 8
Y[1]= 3
Y[2]= 7
Y[3]= 7
Y[4]= 5
Y[5]= 3
Y[6]= 3
Y[7]= 5
Y[8]= 4
Y[9]= 4
```

Рисунок 4.7 – Результати розв’язання задачі 4.1 (спосіб 4А)

Пояснення: Для того щоб задати елементи масиву цілого типу за допомогою генератора випадкових чисел, наприклад, у діапазоні від 1 до 9, використаємо оператор:

$$Y[i] = \text{rnd.Next}(1,10).$$

Для задавання дійсних елементів масиву за допомогою генератора випадкових чисел можна використовувати оператор:

$$Y[i] = \text{rnd.NextDouble}()$$

За допомогою цього оператора будуть згенеровані випадкові дійсні числа в діапазоні від 0 до 1.

Спосіб 4В. Програма генерує і виводить із точністю до трьох знаків дійсні числа в діапазоні $-5,0..5,0$. Величина зазначеного діапазону дорівнює 10,0 ($MaxVal - MinVal = 5 - (-5) = 10$), тому коефіцієнт масштабування дорівнює десяти. Коефіцієнт зсуву дорівнює $-5,0$.

Програмний код:

```
using System;

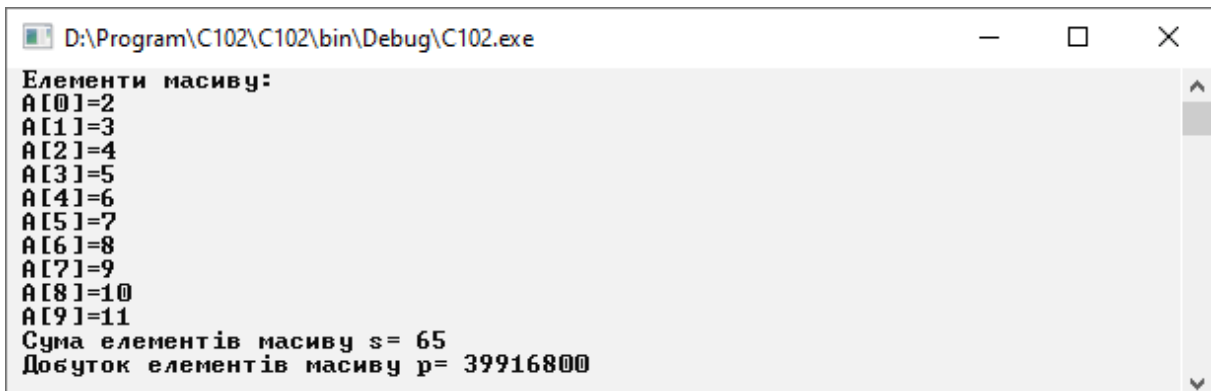
namespace C105
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву
            double[] Y = new double[n];
            int i;
            Random rnd = new Random();
            double Scale = 10.0, Shift = -5;
            Console.WriteLine(" Елементи масиву: ");
            for (i = 0; i < n; i++)
            {
```


Програмний код:

```
using System;

namespace C102
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] A = new int[10];
            int i; double s, p;
            // Обчислення елементів масиву і виведення їх на екран
            Console.WriteLine(" Елементи масиву: ");
            for (i = 0; i < 10; i++)
            {
                A[i] = i + 2;
                Console.WriteLine($" A[{i}]={A[i]}");
            }
            // Накопичення суми і добутку елементів масиву
            s = 0; p = 1.0;
            for (i = 0; i < 10; i++)
            {
                s = s + A[i];
                p = p * A[i];
            }
            Console.WriteLine($" Сума елементів масиву s= {s}");
            Console.WriteLine($" Добуток елементів масиву p= {p}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.11.



```
D:\Program\C102\C102\bin\Debug\C102.exe
Елементи масиву:
A[0]=2
A[1]=3
A[2]=4
A[3]=5
A[4]=6
A[5]=7
A[6]=8
A[7]=9
A[8]=10
A[9]=11
Сума елементів масиву s= 65
Добуток елементів масиву p= 39916800
```

Рисунок 4.11 – Результати розв’язання задачі 4.2

Задача 4.3. Створити одновимірний масив $A(7)$, елементи масиву обчислити за формулою. Для накопичення суми всіх елементів масиву використовувати оператор циклу `foreach`.

Програмний код:

```
using System;

namespace C45
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 7;
            int[] A = new int[n];
            int i;
            int sum = 0;
            // Заповнення масиву значеннями і виведення на екран
            Console.WriteLine(" Масив A: ");
            for (i = 0; i < n; i++)
            {
                A[i] = 2 * i + 7;
                Console.Write(" " + A[i]);
            }
            Console.WriteLine("\n");
            foreach (int k in A) sum = sum + k;
            Console.WriteLine($" Сума елементів масиву sum = {sum}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.12.



```
D:\Program\C45\C45\bin\Debug\C45.exe
Масив A:
7 9 11 13 15 17 19
Сума елементів масиву sum = 91
```

Рисунок 4.12 – Результати розв’язання задачі 4.3

Пояснення:

1. `Console.WriteLine("\n");`

Рядкова константа "\n" містить керуючу послідовність символів \n – переведення рядка.

2. `foreach (int k in A) sum = sum + k;`

Оператор `foreach` служить для циклічного звернення до елементів *колекції*, що представляє собою групу об'єктів. Оператор циклу `foreach` діє таким чином. Коли цикл починається, перший елемент масиву вибирається і присвоюється змінній циклу. На кожному наступному кроці ітерації вибирається наступний елемент масиву, який зберігається в змінній циклу. Цикл завершується, коли всі елементи масиву виявляться обраними. Отже, оператор `foreach` циклічно опитує масив по окремих його елементах від початку і до кінця.

Слід, однак, мати на увазі, що змінна циклу в операторі `foreach` служить тільки для читання. Це означає, що привласнюючи цієї змінній нове значення, не можна змінити вміст масиву.

Задача 4.4.

а) Створити одновимірний масив $A(12)$. Елементи масиву a_i обчислюються за формулою:

$$a_i = 12 + i/\sin(i + 1); \quad i = 0,1,\dots,11.$$

б) Обчислити суму елементів масиву з n -го до k -го (при $n = 5; k = 7$).

Обчислення суми здійснити за допомогою операторів циклу FOR, WHILE або DO.

с) Вивести елементи масиву:

- у зворотному порядку(за убутанням індексу) з k -го до n -го;
- парні (непарні) за індексом або кожний n -й елемент масиву.

Створити консольний додаток.

Програмний код:

```
using System;
```

```
namespace C37
```

```
{
```

```
    class Program
```

```
    {
```

```

static void Main(string[] args)
{
    const int n = 12; // кількість елементів масиву
    double[] A = new double[n];
    int i;
    double s;
    // Обчислення і виведення елементів масиву
    Console.WriteLine(" Масив A: ");
    for (i = 0; i < n; i++)
    {
        A[i] = 12 + i / Math.Sin(i + 1);
        Console.WriteLine($" A[{i}]= {A[i]:F3}");
    }
    Console.WriteLine();
    // Обчислення суми елементів масиву з 5-го до 7-го
    Console.WriteLine(" Сума елементів масиву з 5-го до 7-го");
    s = 0;
    for (i = 5; i <= 7; i++)
        s = s + A[i];
    Console.WriteLine($" s= {s:F3}");
    Console.WriteLine();
    //Виведення елементів масиву в зворотному порядку з 7-го до 5-го
    Console.WriteLine(" Виведення елементів масиву в зворотному
        порядку з 7-го до 5-го " );
    for (i = 7; i >= 5; i--)
        Console.WriteLine($" A[{i}]= {A[i]:F3}");
    Console.WriteLine();
    // Виведення парних за індексом елементів масиву
    Console.WriteLine(" Виведення парних за індексом елементів
        масиву");
    for (i = 2; i <= 10; i = i + 2)
        Console.WriteLine($" A[{i}]= {A[i]:F3}");
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 4.13.

```

D:\Program\C37\C37\bin\Debug\C37.exe
Масив А:
A[0]= 12,000
A[1]= 13,100
A[2]= 26,172
A[3]= 8,036
A[4]= 7,829
A[5]= -5,894
A[6]= 21,133
A[7]= 19,075
A[8]= 31,412
A[9]= -4,543
A[10]= 2,000
A[11]= -8,500

Сума елементів масиву з 5-го до 7-го
s= 34,313

Виведення елементів масиву в зворотному порядку з 7-го до 5-го
A[7]= 19,075
A[6]= 21,133
A[5]= -5,894

Виведення парних за індексом елементів масиву
A[2]= 26,172
A[4]= 7,829
A[6]= 21,133
A[8]= 31,412
A[10]= 2,000

```

Рисунок 4.13 – Результати розв’язання задачі 4.4

Таблиця 4.1 – Варіанти завдань

Вар.	Вираз для обчислення елементів масиву	Значення n і k
1	2	3
1	$a_i = 1/2^i + 1/3^i; \quad i = 0,1,\dots,11$	$n = 3; k = 5$
2	$a_i = (2 \cdot i - 1) \cdot 2^i; \quad i = 0,1,\dots,11$	$n = 2; k = 4$
3	$a_i = 1/(3i - 2) \cdot (3i + 1); \quad i = 0,1,\dots,11$	$n = 4; k = 6$
4	$a_i = 10^i/(i + 1); \quad i = 0,1,\dots,11$	$n = 5; k = 8$
5	$a_i = 10/i^i; \quad i = 0,1,\dots,11$	$n = 2; k = 5$
6	$a_i = \ln(i + 2) + i^2; \quad i = 0,1,\dots,11$	$n = 3; k = 6$
7	$a_i = i^{\ln(i+1)} / (\ln(5 \cdot i))^i; \quad i = 0,1,\dots,11$	$n = 2; k = 6$
8	$a_i = 10/(i + 3); \quad i = 0,1,\dots,11$	$n = 3; k = 7$
9	$a_i = e^i/(i + 5); \quad i = 0,1,\dots,11$	$n = 2; k = 8$
10	$a_i = (i + 3)^2 \cdot e^i; \quad i = 0,1,\dots,11$	$n = 3; k = 6$

Задача 4.5. Програма визначає суму і кількість від’ємних елементів одновимірного масиву.

Програмний код:

```
using System;

namespace C45a
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 10;
            int i;
            int[] A = new int[n] {1, -4, -7, 8, -10, 12, -14, -17,
                                20, 21 };

            Console.WriteLine(" Масив A: ");
            for (i = 0; i < n; i++)
                Console.Write(" " + A[i]);
            Console.WriteLine();
            long sum = 0; // сума від'ємних елементів
            long num = 0; // кількість від'ємних елементів
            for (i = 0; i < n; i++)
                if (A[i] < 0)
                {
                    sum = sum + A[i];
                    num = num + 1;
                }
            Console.WriteLine($" Сума від'ємних елементів = {sum}");
            Console.WriteLine($" Кількість від'ємних елементів = {num}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.14.



```
D:\Program\C45a\C45a\bin\Debug\C45a.exe
Масив А:
1 -4 -7 8 -10 12 -14 -17 20 21
Сума від'ємних елементів = -52
Кількість від'ємних елементів = 5
```

Рисунок 4.14 – Результати розв'язання задачі 4.5

Задача 4.6. Обчислити $z = a \sum_{i=4}^{10} x_i + 3 \sqrt[3]{\prod_{i=2}^8 (y_i + b)}$,

де $a = 2,34$; $b = 5,72$;

x_i – елементи одновимірного масиву $X(12)$; $i = 0, 1, \dots, 11$;

y_i – елементи одновимірного масиву $Y(10)$; $i = 0, 1, \dots, 9$;

$\sum_{i=4}^{10} x_i$ – сума елементів одновимірного масиву X з 4-го до 10-го.

$\prod_{i=2}^8 (y_i + b)$ – добуток виразу $(y_i + b)$, де y_i – елементи одновимірного

масиву Y з 2-го до 8-го.

Значення елементів масивів обчислювати за формулами:

$$x_i = (i + 12, 0) / 7, 0; \quad y_i = (i + 4, 0) / 5, 0.$$

Програмний код:

```
using System;

namespace C21a
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву X
            const int m = 10; // кількість елементів масиву Y
            int i; double z, s, p;
            double[] X = new double[n];
            double[] Y = new double[m];
            double a = 2.34, b = 5.72;
            // Обчислення і виведення елементів масиву X
            Console.WriteLine(" Масив X ");
            for (i = 0; i < n; i++)
            {
                X[i] = (i + 12.0) / 7.0;
                Console.WriteLine($" X[{i}] = {X[i]:F2}");
            }
            Console.WriteLine();
            // Обчислення і виведення елементів масиву Y
            Console.WriteLine(" Масив Y ");
            for (i = 0; i < m; i++)
            {
                Y[i] = (i + 4.0) / 5.0;
                Console.WriteLine($" Y[{i}] = {Y[i]:F2}");
            }
        }
    }
}
```

```

Console.WriteLine();
// Обчислення суми елементів масиву X з 4-го до 10-го
s = 0;
for (i = 4; i <= 10; i++)
    s = s + X[i];
// Обчислення добутку елементів масиву Y з 2-го до 8-го
p = 1;
for (i = 2; i <= 8; i++)
    p = p * (Y[i] + b);
z = a * s + Math.Pow(p, (1.0 / 3));
Console.WriteLine($"z= {z:F3}");
Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 4.15.

```

D:\Program\C21a\C21a\bin\Debug\C21a.exe
Масив X
X[0]= 1,71
X[1]= 1,86
X[2]= 2,00
X[3]= 2,14
X[4]= 2,29
X[5]= 2,43
X[6]= 2,57
X[7]= 2,71
X[8]= 2,86
X[9]= 3,00
X[10]= 3,14
X[11]= 3,29

Масив Y
Y[0]= 0,80
Y[1]= 1,00
Y[2]= 1,20
Y[3]= 1,40
Y[4]= 1,60
Y[5]= 1,80
Y[6]= 2,00
Y[7]= 2,20
Y[8]= 2,40
Y[9]= 2,60

z= 154,886

```

Рисунок 4.15 – Результати розв’язання задачі 4.6

Задача 4.7. Відсортувати елементи одновимірного масиву.

Для сортування елементів одновимірного масиву існує

`Array.Sort(a)` – метод `Sort` класу `Array`.

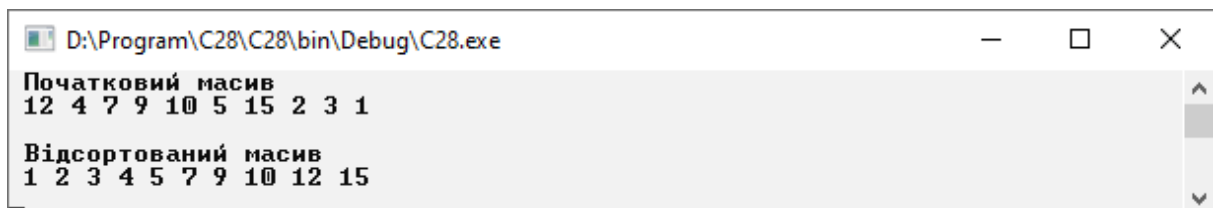
Приклад показує сортування масиву цілих чисел `A(10)`.

Програмний код:

```
using System;

namespace C28
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 10; // кількість елементів масиву
            int i;
            int[] A = new int[n] { 12, 4, 7, 9, 10, 5, 15, 2, 3, 1};
            Console.WriteLine(" Початковий масив");
            // Виведення елементів початкового масиву
            for (i = 0; i < n; i++)
                Console.Write(" " + A[i]);
            Console.WriteLine();
            Array.Sort(A);
            Console.WriteLine();
            Console.WriteLine(" Відсортований масив");
            // Виведення елементів відсортованого масиву
            for (i = 0; i < n; i++)
                Console.Write(" " + A[i]);
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.16.



```
D:\Program\C28\C28\bin\Debug\C28.exe
Початковий масив
12 4 7 9 10 5 15 2 3 1
Відсортований масив
1 2 3 4 5 7 9 10 12 15
```

Рисунок 4.16 – Результати розв’язання задачі 4.7

Задача 4.8. Створити консольний додаток для розрахування середнього арифметичного елементів одновимірного масиву $A(n)$, де n –

кількість елементів масиву. Значення n задати довільно, елементи масиву створити за допомогою генератора випадкових чисел.

Програмний код:

```
using System;

namespace C30
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;    // кількість елементів масиву
            int i;
            double s; // сума елементів масиву
            double sr; // середнє арифметичне елементів масиву
            Console.WriteLine(" Введіть кількість елементів масиву =? ");
            n = Convert.ToInt32(Console.ReadLine());
            int[] A = new int[n];
            Random rnd = new Random();
            Console.WriteLine(" Елементи масиву: ");
            s = 0;
            for (i = 0; i < n; i++)
            {
                A[i] = rnd.Next(1, 20);
                Console.WriteLine($" A[{i}]={A[i]}");
                s = s + A[i];
            }
            sr = s / n;
            Console.WriteLine($" Середнє арифметичне елементів
                               масиву = {sr:F3}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.17.

```

D:\Program\C30\C30\bin\Debug\C30.exe
Введіть кількість елементів масиву =? 15
Елементи масиву:
A[0]=15
A[1]=11
A[2]=6
A[3]=6
A[4]=17
A[5]=6
A[6]=13
A[7]=15
A[8]=14
A[9]=16
A[10]=9
A[11]=12
A[12]=16
A[13]=7
A[14]=16
Середнє арифметичне елементів масиву = 11,933

```

Рисунок 4.17 – Результати розв’язання задачі 4.8

Задача 4.9. Створити консольний додаток для розрахування максимального елемента одновимірного масиву дійсних чисел $A(12)$ і його індексу (порядкового номера).

Блок-схема знаходження максимального елемента одновимірного масиву та його індексу наведена на рис. 4.18.

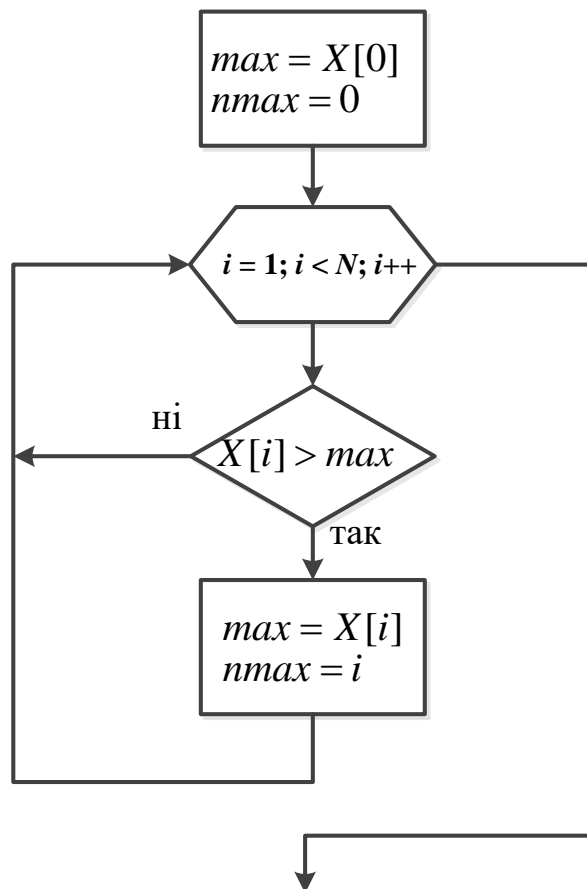


Рисунок 4.18 – Блок-схема знаходження максимального елемента одновимірного масиву та його індексу

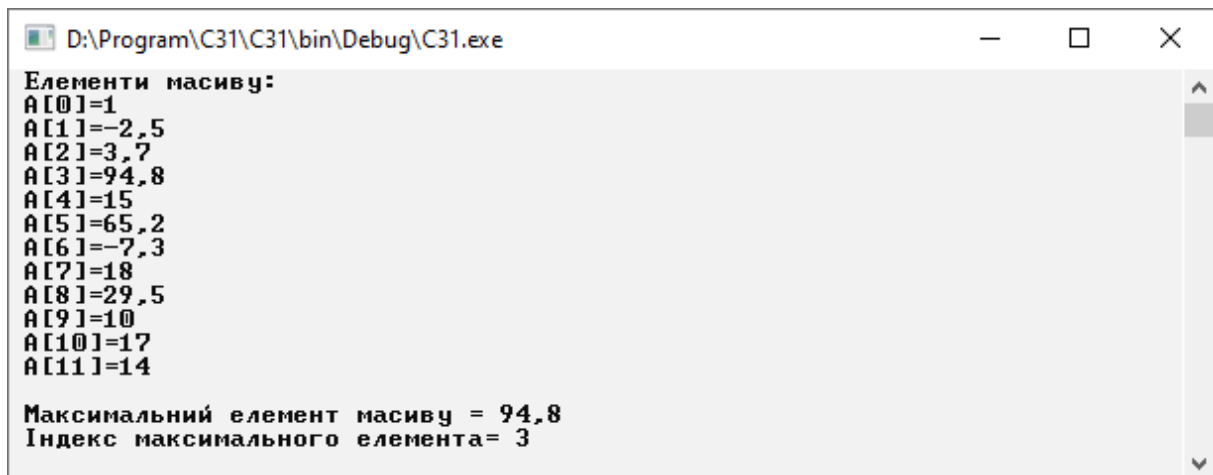
Пояснення: Алгоритм розв'язання задачі такий. Нехай змінна на ім'я **max** зберігає значення максимального елемента масиву, а змінна на ім'я **nmax** – його номер. Припустимо, що перший елемент масиву (елемент з індексом 0) є максимальним, і запишемо його у змінну **max**, а у змінну **nmax** – його номер (тобто 0). Далі всі елементи, починаючи з індексу 1, порівнюємо у циклі з максимальним. Якщо поточний елемент масиву виявиться більше за максимальний, то записуємо його у змінну **max**, а у змінну **nmax** – поточне значення індексу **i**.

Програмний код:

```
using System;

namespace C31
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву
            double max; // максимальний елемент
            int imax; // індекс максимального елемента
            int i;
            double[] A = new double[n] { 1.0, -2.5, 3.7, 94.8, 15.0,
                65.2, -7.3, 18.0, 29.5, 10.0, 17.0, 14.0 };
            Console.WriteLine(" Елементи масиву: ");
            // Виведення елементів масиву
            for (i = 0; i < n; i++)
                Console.WriteLine($" A[{i}]={A[i]}");
            max = A[0]; imax = 0;
            for (i = 1; i < n; i++)
            {
                if (A[i] > max)
                {
                    max = A[i];
                    imax = i;
                }
            }
            Console.WriteLine();
            Console.WriteLine($" Максимальний елемент масиву = {max}");
            Console.WriteLine($" Індекс максимального елемента= {imax}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.19.



```
D:\Program\C31\C31\bin\Debug\C31.exe
Елементи масиву:
A[0]=1
A[1]=-2,5
A[2]=3,7
A[3]=94,8
A[4]=15
A[5]=65,2
A[6]=-7,3
A[7]=18
A[8]=29,5
A[9]=10
A[10]=17
A[11]=14

Максимальний елемент масиву = 94,8
Індекс максимального елемента= 3
```

Рисунок 4.19 – Результати розв’язання задачі 4.9

Задача 4.10. В одновимірному масиві цілих чисел $A(12)$ всі додатні числа збільшити вдвічі, а також обчислити кількість чисел, що дорівнюють 0, і суму всіх від’ємних чисел. Блок-схему розв’язання наведено на рис. 4.21.

Програмний код:

```
using System;

namespace C32
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву
            int i, k; double s;
            int[] A = new int[n] { 5, 8, 0, 14, -15, 17, 0, -20, 0,
                                   23, 25, -30 };

            // Виведення елементів масиву
            Console.WriteLine(" Початковий масив A: ");
            for (i = 0; i < n; i++)
                Console.Write(" " + A[i]);
            Console.WriteLine("\n");
            k = 0; s = 0;
```

```

for (i = 0; i < n; i++)
{
    if (A[i] > 0)
        A[i] = 2 * A[i];
    else if (A[i] == 0)
        k = k + 1;
    else
        s = s + A[i];
}
// Виведення елементів нового масиву
Console.WriteLine( " Новий масив A: ");
for (i = 0; i < n; i++)
    Console.Write(" " + A[i]);
Console.WriteLine("\n");
Console.WriteLine($" Кількість елементів рівних нулю =
                                     {k}");
Console.WriteLine($" Сума всіх від'ємних елементів =
                                     {s}");

Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 4.20.

```

D:\Program\C32\C32\bin\Debug\C32.exe
Початковий масив A:
5 8 0 14 -15 17 0 -20 0 23 25 -30

Новий масив A:
10 16 0 28 -15 34 0 -20 0 46 50 -30

Кількість елементів рівних нулю = 3
Сума всіх від'ємних елементів = -65

```

Рисунок 4.20 – Результати розв'язання задачі 4.10

Пояснення:

```
Console.WriteLine("\n");
```

Рядкова константа "\n" містить керуючу послідовність символів \n – переведення рядка.

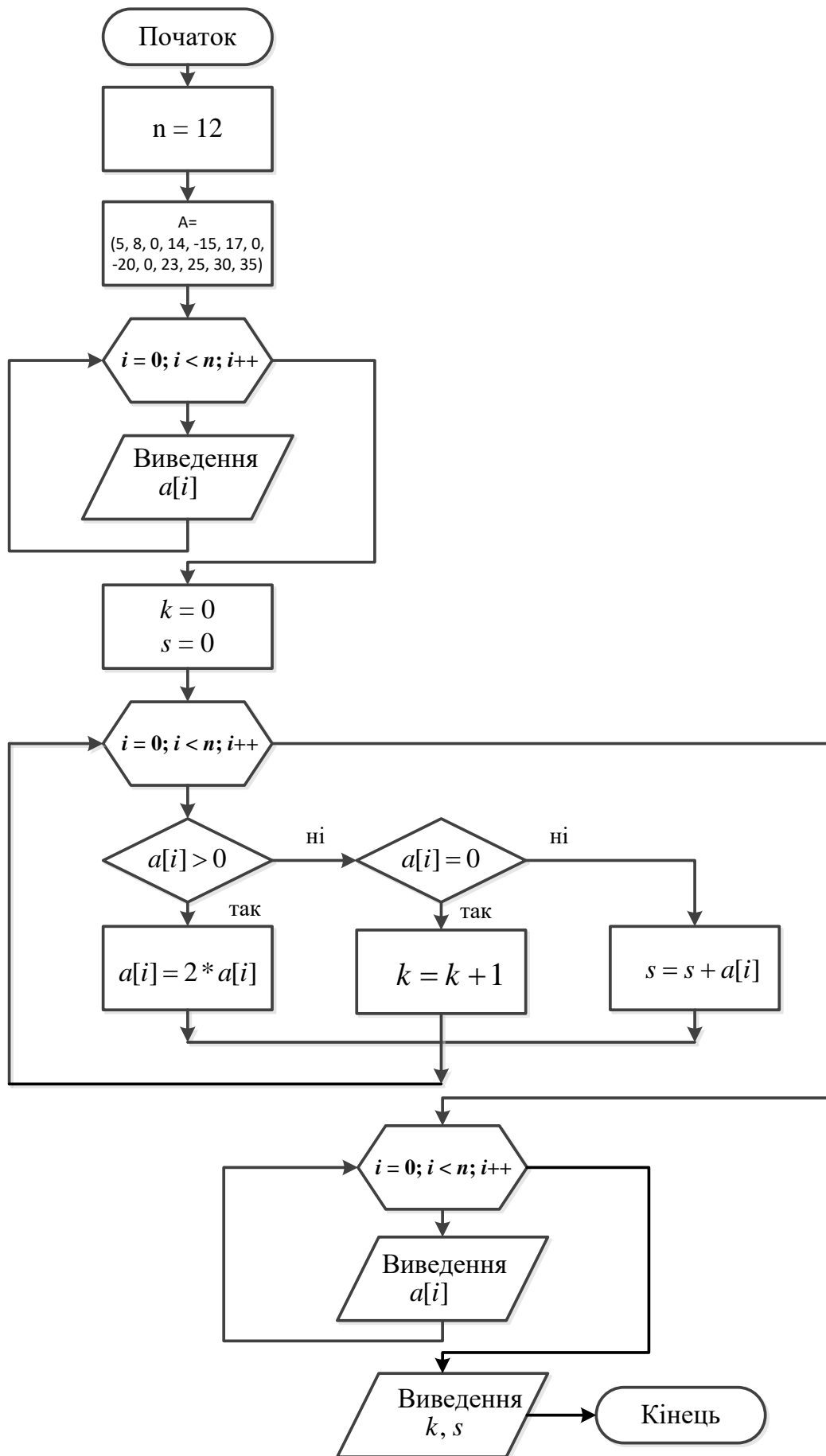


Рисунок 4.21 – Блок-схема алгоритму до задачі 4.10

Задача 4.11. В одновимірному масиві цілих чисел $A(12)$ визначити суму елементів, кратних трьом, та їх кількість.

Програмний код:

```
using System;

namespace C33
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 12; // кількість елементів масиву
            int i, k; double s;
            int[] A = new int[n] { 2, 3, 7, 9, 11, 12, 14, 15, 16,
                                   17, 18, 20 };

            // Виведення елементів масиву
            Console.WriteLine( " Масив A: ");
            for (i = 0; i < n; i++)
                Console.Write(" " + A[i]);
            Console.WriteLine("\n");
            k = 0; s = 0;
            Console.WriteLine(" Елементи кратні трьом: ");
            for (i = 0; i < n; i++)
            {
                if (A[i] % 3 == 0)
                {
                    k = k + 1;
                    s = s + A[i];
                    Console.Write(" " + A[i]);
                }
            }
            Console.WriteLine("\n");
            Console.WriteLine($" Кількість елементів кратних трьом =
                                   {k}");
            Console.WriteLine($" Сума елементів кратних трьом =
                                   {s}");

            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.22.

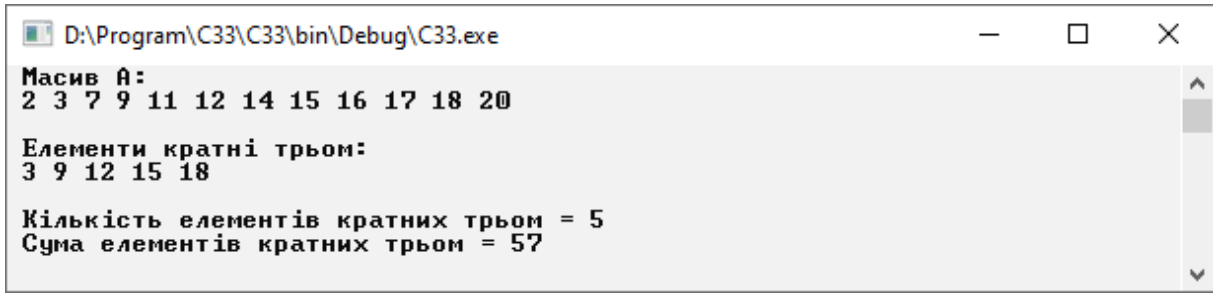


Рисунок 4.22 – Результати розв’язання задачі 4.11

Задача 4.12. В одновимірному масиві цілих чисел $A(15)$ обчислити суму елементів, розташованих після першого від’ємного елемента.

Елементи масиву обчислити за формулою: $A_i = (i - 9, 0) / (i - 3, 2)$.

Блок-схему розв’язання задачі наведено на рис. 4.23.

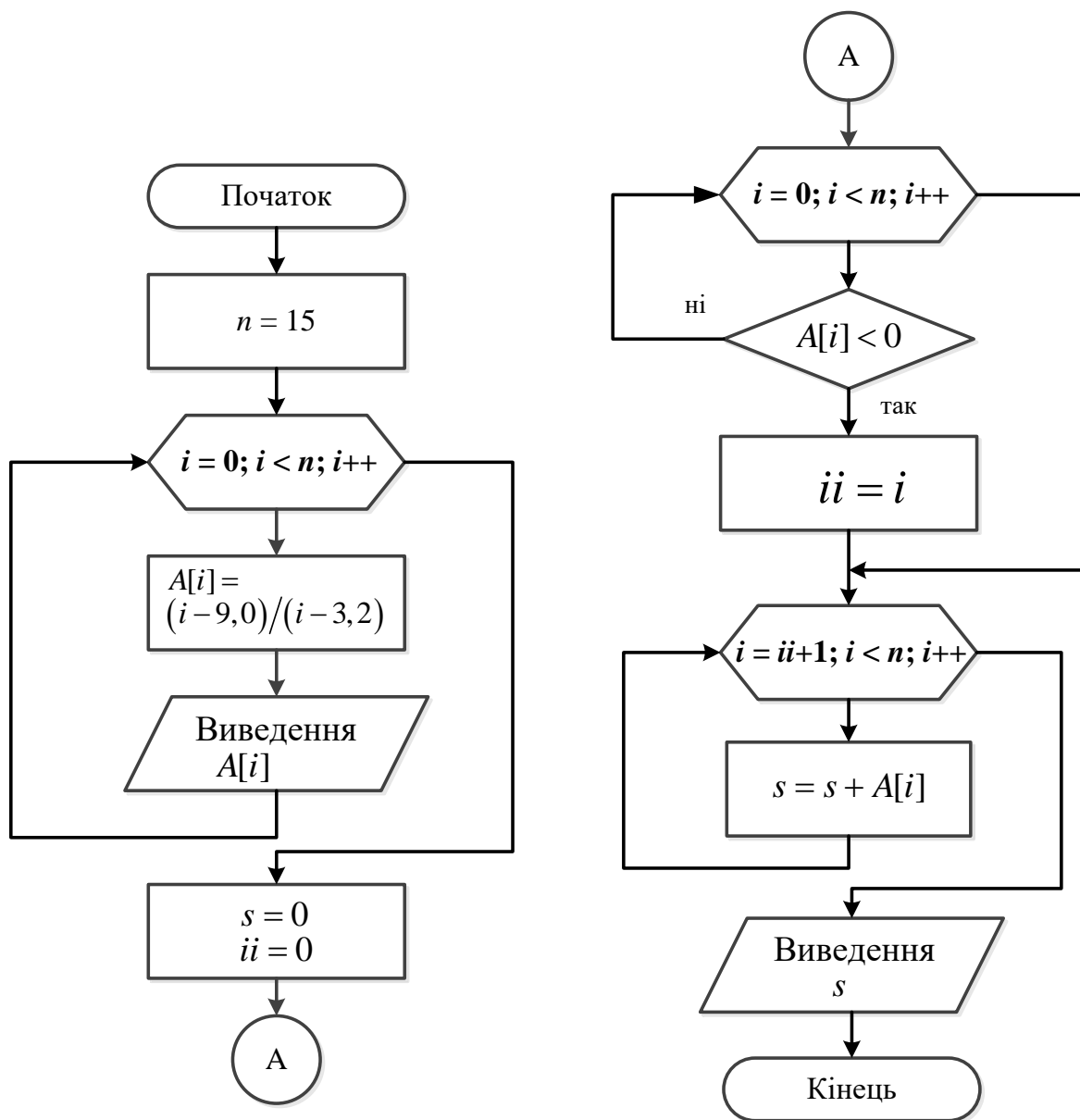


Рисунок 4.23 – Блок-схема алгоритму до задачі 4.12

Програмний код:

```
using System;

namespace C303
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 15; // кількість елементів масиву
            double[] A = new double[n];
            int i;
            int ii = 1000; //Індекс першого від'ємного елемента
            Console.WriteLine(" Елементи масиву: ");
            for (i = 0; i < n; i++)
            {
                A[i] = (i - 9.0) / (i - 3.2);
                Console.WriteLine($" A[{i}]= {A[i]:F3}");
            }
            for (i = 0; i < n; i++)
            {
                if (A[i] < 0)
                {
                    ii = i;
                    break;
                }
            }
            double s = 0;
            for (i = ii + 1; i < n; i++)
            {
                s = s + A[i];
            }
            Console.WriteLine($" Сума елементів s= {s:F3}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 4.24.

```
D:\Program\C303\C303\bin\Debug\C303.exe
Елементи масиву:
A[0]= 2,813
A[1]= 3,636
A[2]= 5,833
A[3]= 30,000
A[4]= -6,250
A[5]= -2,222
A[6]= -1,071
A[7]= -0,526
A[8]= -0,208
A[9]= 0,000
A[10]= 0,147
A[11]= 0,256
A[12]= 0,341
A[13]= 0,408
A[14]= 0,463
Сума елементів s= -2,413
```

Рисунок 4.24 – Результати розв’язання задачі 4.12

4.2 Завдання для виконання лабораторних робіт (частина 1)

Задача 1. Обчислити значення виразу за формулою

$$y = \sum_{i=0}^{20} a_i b_i,$$

де a_i – елементи одновимірного масиву $A(21)$; $i = 0, 1, \dots, 20$;

b_i – елементи одновимірного масиву $B(21)$; $i = 0, 1, \dots, 20$.

Задача 2. Обчислити елементи масиву z_j :

$$z_j = (x_j + y_j) / p,$$

де: $j = 0, 1, \dots, 9$; $p = 23$;

x_j – елементи одновимірного масиву $X(10)$;

y_j – елементи одновимірного масиву $Y(10)$;

z_j – елементи одновимірного масиву $Z(10)$.

Задача 3. Обчислити значення виразу

$$Y = \prod_{i=0}^9 (a_i - 10) / 5,$$

де a_i – елементи одновимірного масиву $A(10)$; $i = 0, 1, \dots, 9$.

Задача 4. Обчислити значення виразу

$$P = \sum_{i=0}^9 a_i - \sum_{i=0}^9 b_i,$$

де a_i – елементи одновимірного масиву $A(10)$; $i = 0, 1, \dots, 9$;

b_i – елементи одновимірного масиву $B(10)$; $i = 0, 1, \dots, 9$.

Задача 5. Обчислити суму всіх непарних за індексом елементів масиву $A(30)$.

Задача 6. Обчислити значення виразу

$$d = \sum_{i=0}^8 (a_i + b_i)^{1/3},$$

де a_i – елементи одновимірного масиву $A(9)$; $i = 0, 1, \dots, 8$;

b_i – елементи одновимірного масиву $B(9)$; $i = 0, 1, \dots, 8$.

Задача 7. Обчислити значення виразу

$$Z = \left\{ \sum_{j=0}^{20} (y_j + x_j) \right\}^{1/2},$$

де x_j – елементи одновимірного масиву $X(21)$; $i = 0, 1, \dots, 20$;

y_j – елементи одновимірного масиву $Y(21)$; $i = 0, 1, \dots, 20$.

Задача 8. Знайти суму перших п'ятнадцяти елементів масиву $B(40)$.

Задача 9. Знайти добуток останніх 10 елементів масиву $X(30)$.

Задача 10. Обчислити значення виразу

$$R = \sum_{i=5}^{20} (\cos x_i \cdot \sin y_i),$$

де x_i – елементи одновимірного масиву $X(21)$; $i = 0, 1, \dots, 20$;

y_i – елементи одновимірного масиву $Y(21)$; $i = 0, 1, \dots, 20$.

Задача 11. Обчислити значення виразу

$$X = \sqrt[3]{\sum_{i=2}^{15} (z_i + a)},$$

де $a = 4,65$; z_i – елементи одновимірного масиву $Z(16)$; $i = 0,1,\dots,15$.

Задача 12. Обчислити значення виразу

$$F1 = c^2 \prod_{i=2}^8 (y_i \cdot x)^2,$$

де $c = 12$; $x = 1,5$; y_i – елементи одновимірного масиву $Y(9)$; $i = 0,1,\dots,8$.

Задача 13. Обчислити суму всіх парних за індексом елементів масиву $A(20)$, $i = 0,1,\dots,19$.

Задача 14. Обчислити суму всіх непарних за індексом елементів масиву $A(15)$, $i = 0,1,\dots,14$.

4.3 Завдання для виконання лабораторних робіт (частина 2)

Задача 1. Обчислити добуток додатних елементів масиву $A(15)$.

Задача 2. Обчислити суму від'ємних елементів масиву $B(20)$.

Задача 3. В одновимірному масиві $D(20)$ обчислити кількість елементів, що дорівнюють нулю, і визначити їх індекси.

Задача 4. В одновимірному масиві $C(15)$ обчислити кількість додатних елементів.

Задача 5. В одновимірному масиві $D(20)$ усі елементи, що дорівнюють нулю, замінити на число 10.

Задача 6. Обчислити середнє арифметичне значення елементів одновимірного масиву $A(15)$.

Задача 7. Визначити максимальний елемент одновимірного масиву $B(15)$ та його порядковий номер.

Задача 8. Обчислити мінімальний елемент одновимірного масиву $C(15)$ та його порядковий номер.

Задача 9. Визначити суму елементів масиву $N(20)$, що кратні трьом.

Задача 10. Переписати підряд у масив Y додатні та у масив Z від'ємні елементи масиву $X(20)$.

Задача 11. В одновимірному масиві $D(20)$ максимальний елемент замінити на число 100.

Задача 12. В одновимірному масиві $M(20)$ мінімальний елемент замінити на число 200.

Задача 13. Знайти максимальний та мінімальний елементи одновимірного масиву $D(20)$ і поміняти їх місцями.

Задача 14. В одновимірному масиві $A(20)$ визначити суму і кількість елементів, які менші -5 .

Задача 15. В одновимірному масиві $B(20)$ знайти найбільший елемент серед від'ємних чисел.

Задача 16. В одновимірному масиві $X(15)$ знайти середнє арифметичне значення додатних елементів масиву.

Задача 17. Написати програму, в якій вивести з точністю до чотирьох знаків частку від ділення суми і добутку ($\sum a_i / \prod a_i$) всіх елементів цілого масиву з п'яти елементів, що заповнений випадковими числами в діапазоні 1..9.

Тема 5. Двовимірні масиви

Мета роботи: вивчити оператори оголошення двовимірних масивів, способи наповнення масивів, введення-виведення елементів масивів, накопичення суми і добутку елементів масивів.

5.1 Приклади розв'язання задач

Задача 5.1. Створити двовимірний масив A розміром 3×4 , де 3 – кількість рядків; 4 – кількість стовпців масиву

$$A = (a_{ij}; i = 0,1,\dots,2; j = 0,1,\dots,3).$$

Нижній індекс масиву за 1-м і 2-м вимірами = 0,

верхній індекс масиву за 1-м виміром =2,

верхній індекс масиву за 2-м виміром =3.

Здійснити уведення і виведення елементів двовимірного масиву.

Спосіб 1. Елементи масиву задамо у програмному коді.

Програмний код:

```
using System;

namespace C144
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 3;        // кількість рядків масиву A
            const int m = 4;        // кількість стовпців масиву A
            int i, j;
            double[,] A = new double[n, m]
                {{1,2,3,4},
                {5,6,7,8},
                {9,10,11,12}};

            // Виведення елементів масиву
            Console.WriteLine("    Масив A ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                {
                    Console.Write($"{A[i,j],8:F2} ");
                }
            }
        }
    }
}
```

```

        }
        Console.WriteLine();
    }
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.1.

Масив А			
1,00	2,00	3,00	4,00
5,00	6,00	7,00	8,00
9,00	10,00	11,00	12,00

Рисунок 5.1 – Результати розв’язання задачі 5.1 (спосіб 1)

Пояснення:

```

1) double[,] A = new double[n, m]
    {{1,2,3,4},
     {5,6,7,8},
     {9,10,11,12}};

```

Одночасно із створенням масиву він заповнюється необхідними значеннями. Якщо зазначено n , m і перелік ініціалізаторів, то n і m повинні бути тільки константами.

Спосіб 2. Елементи масиву обчислимо за формулою:

$$a_{ij} = (i^2 - 5)/(i + j + 1).$$

Програмний код:

```

using System;

namespace C142
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 3; // кількість рядків масиву А
            const int m = 4; // кількість стовпців масиву А
            int i, j;
            double[,] A = new double[n, m];
            // Обчислення елементів масиву за формулою
            // та виведення їх на екран

```

```

Console.WriteLine("  Масив A  ");
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        A[i, j] = (double)(i * i - 5) / (i + j + 1);
        Console.Write($"{A[i,j],9:F3} ");
    }
    Console.WriteLine();
}
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.2.

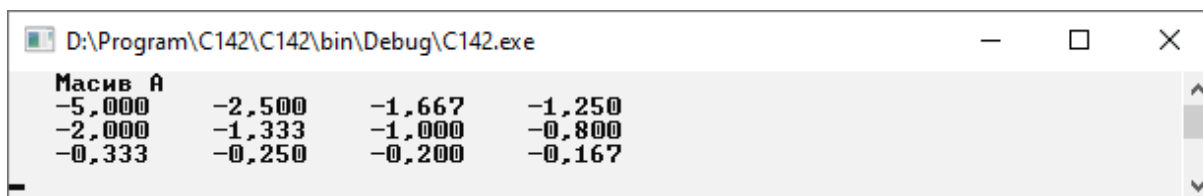


Рисунок 5.2 – Результати розв’язання задачі 5.1 (спосіб 2)

Пояснення:

1) `double[,] A = new double[n,m];`

Оголошується двовимірний масив A із $n \times m$ дійсних чисел типу `double`. Одночасно масив ініціалізується (заповнюється нулями);

n і m повинні бути або константами, або виразами типу, що приводяться до цілого.

2) `A[i,j] = (double)(i * i - 5) / (i + j + 1);`

При діленні цілих чисел остачу буде відкинуто; щоб цього не відбулося, необхідно змінити тип хоча б одного з них.

Спосіб 3. Елементи масиву визначити за допомогою генератора випадкових чисел.

Програмний код:

```

using System;

namespace C143
{
    class Program
    {

```

```

static void Main(string[] args)
{
    const int n = 3;        // кількість рядків масиву A
    const int m = 4;        // кількість стовпців масиву A
    int i, j;
    double[,] A = new double[n, m];
    //Створюється екземпляр rnd класу Random
    Random rnd = new Random();
    //Заповнення елементів масиву випадковими числами
    //i виведення їх на екран
    Console.WriteLine("  Масив A  ");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            A[i, j] = rnd.NextDouble() * 10;
            Console.Write($"{A[i,j],8:F2} ");
        }
        Console.WriteLine();
    }
    Console.ReadLine();
}
}
}
}
}

```

Результати розрахунку подано на рис. 5.3.

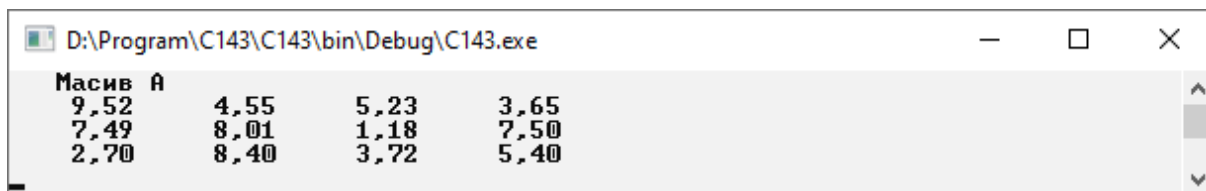


Рисунок 5.3 – Результати розв’язання задачі 5.1 (спосіб 3)

Спосіб 4. Елементи масиву уведемо з клавіатури. Уведення здійснюється по рядках.

Програмний код:

```

using System;

namespace C145
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 3;        // кількість рядків масиву A

```

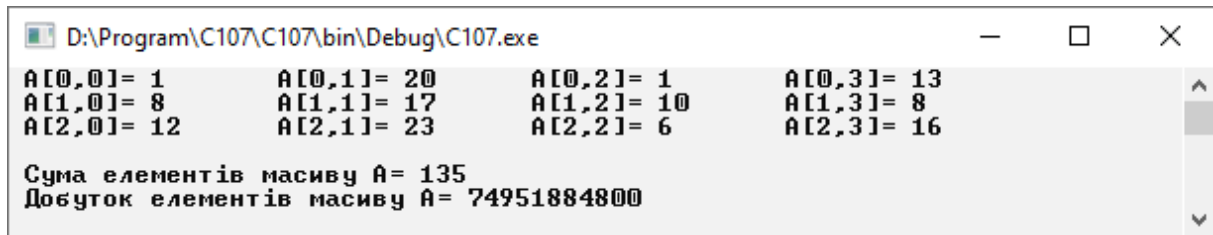

Задача 5.2. Обчислити суму і добуток елементів двовимірного масиву

Програмний код:

```
using System;

namespace C107
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 3; // кількість рядків масиву A
            const int m = 4; // кількість стовпців масиву A
            int[,] A = new int[n, m];
            Random rnd = new Random();
            int sum = 0;
            double p = 1;
            // Заповнення елементів масиву
            // і накопичення їх суми та добутку
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {
                    A[i, j] = rnd.Next(1, 25);
                    sum = sum + A[i, j];
                    p = p * A[i, j];
                }
            }
            // Виведення елементів масиву
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {
                    Console.Write($" A[{i},{j}] = {A[i,j]}\t");
                }
                Console.WriteLine();
            }
            Console.WriteLine();
            Console.WriteLine($" Сума елементів масиву A = {sum}");
            Console.WriteLine($" Добуток елементів масиву A = {p}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 5.5.



```
D:\Program\C107\C107\bin\Debug\C107.exe
A[0,0]= 1      A[0,1]= 20     A[0,2]= 1      A[0,3]= 13
A[1,0]= 8      A[1,1]= 17     A[1,2]= 10     A[1,3]= 8
A[2,0]= 12     A[2,1]= 23     A[2,2]= 6      A[2,3]= 16

Сума елементів масиву А= 135
Добуток елементів масиву А= 74951884800
```

Рисунок 5.5 – Результати розв’язання задачі 5.2

Пояснення:

```
Console.WriteLine($" A[{{i}},{{j}}]= {A[i,j]}\t");
\t керуючий символ : горизонтальна табуляція.
```

Задача 5.3. Створити двовимірний масив A розміром 4×5 .

$$A = (a_{ij}; i = 0,1,\dots,3; j = 0,1,\dots,4).$$

Елементи масиву задати у програмному коді

$$A = \begin{pmatrix} 3 & 5 & 11 & 8 & -7 \\ 42 & 10 & 85 & 31 & 14 \\ 5 & -2 & 12 & 3 & 15 \\ 80 & 18 & -22 & 35 & 40 \end{pmatrix}.$$

Знайти максимальний елемент двовимірного масиву та його індекси.

Блок-схема знаходження максимального елемента двовимірного масиву та його індексів наведена на рис. 5.7.

Програмний код:

```
using System;

namespace C305
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 4;        // кількість рядків масиву А
            const int m = 5;        // кількість стовпців масиву А
            int i, j;
            double max; // максимальний елемент масиву
            int imax, jmax; // індекси максимального елемента
```

```

double[,] A = new double[n, m]
    {{3, 5, 11, 8, -7},
     {42, 10, 85, 31, 14},
     {5, -2, 12, 3, 15},
     {80, 18, -22, 35, 40}};
// Виведення елементів на екран
Console.WriteLine(" Масив A ");
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        Console.Write($"{A[i,j],9:F3} ");
    }
    Console.WriteLine();
}
Console.WriteLine();
// Знаходження максимального елемента
// масиву та його індексів
max = A[0, 0]; imax = 0; jmax = 0;
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        if (A[i, j] >= max)
        {
            max = A[i, j];
            imax = i;
            jmax = j;
        }
    }
}
Console.WriteLine($" Максимальний елемент масиву=
                    {max:F3}");
Console.WriteLine($" Індеси максимального елемента
                    imax= {imax}; jmax= {jmax}");
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.6.

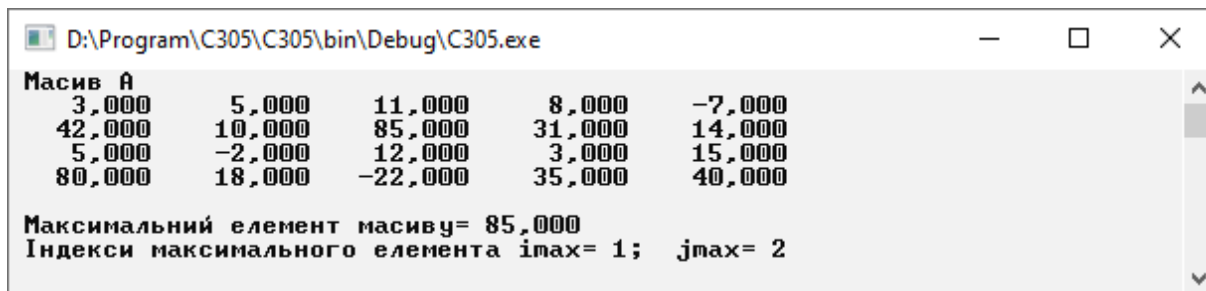


Рисунок 5.6 – Результати розв’язання задачі 5.3

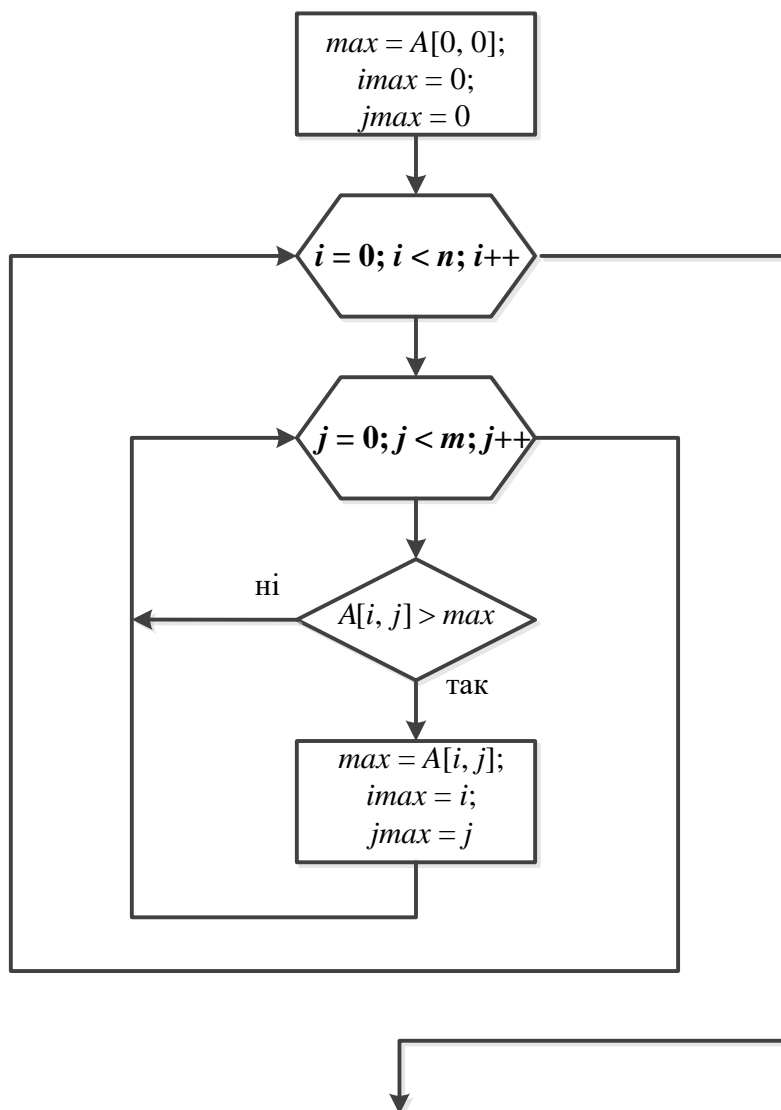


Рис 5.7 – Блок-схема знаходження максимального елемента двовимірного масиву та його індексів

Задача 5.4. Створити двовимірний масив A розміром 4×6 .

Елементи масиву обчислити за формулою: $a_{ij} = (i^2 - 5)/(i + j + 1)$.

Обчислити суму додатних і кількість від’ємних елементів двовимірного масиву A . Блок-схему розв’язання задачі наведено на рис. 5.8.

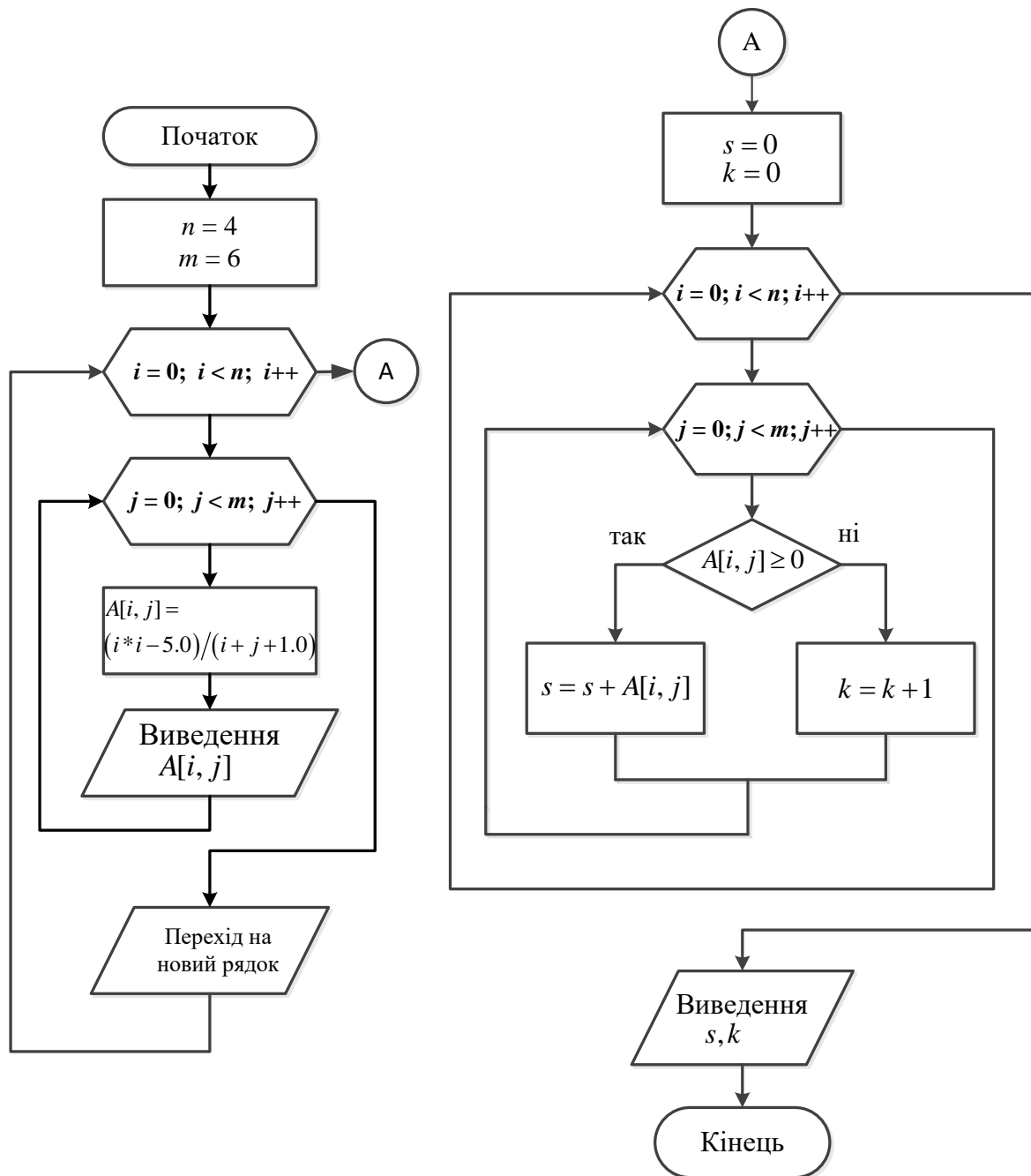


Рисунок 5.8 – Блок-схема алгоритму до задачі 5.4

Програмний код:

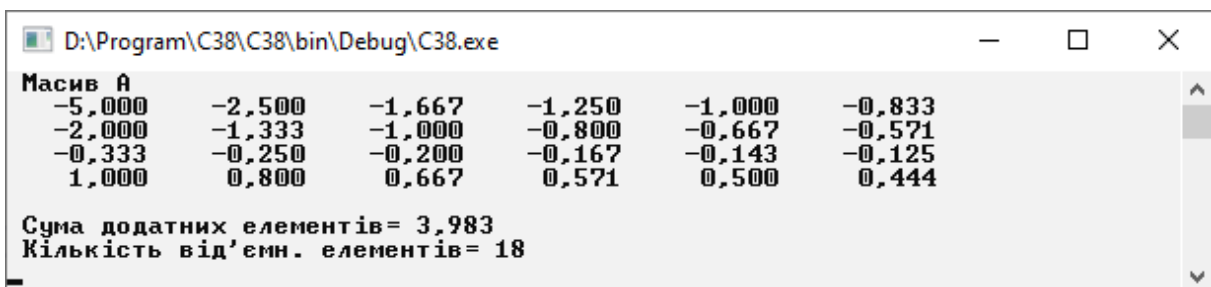
```
using System;
namespace C38
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 4;        // кількість рядків масиву A
            const int m = 6;        // кількість стовпців масиву A
            int i, j, k;
```

```

double s;
double[,] A = new double[n, m];
// Обчислення елементів масиву за формулою
// і виведення їх на екран
Console.WriteLine(" Масив A ");
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        A[i, j] = (i * i - 5.0) / (i + j + 1.0);
        Console.Write($"{A[i,j],9:F3} ");
    }
    Console.WriteLine();
}
Console.WriteLine();
// Обчислення суми додатних та кількості
// від'ємних елементів масиву
s = 0; k = 0;
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        if (A[i, j] >= 0)
            s = s + A[i, j];
        else
            k = k + 1;
    }
}
Console.WriteLine($" Сума додатних елементів= {s:F3}");
Console.WriteLine($" Кількість від'ємн. елементів= {k}");
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.9.



```

D:\Program\C38\C38\bin\Debug\C38.exe
Масив А
-5,000    -2,500    -1,667    -1,250    -1,000    -0,833
-2,000    -1,333    -1,000    -0,800    -0,667    -0,571
-0,333    -0,250    -0,200    -0,167    -0,143    -0,125
 1,000     0,800     0,667     0,571     0,500     0,444

Сума додатних елементів= 3,983
Кількість від'ємн. елементів= 18

```

Рисунок 5.9 – Результати розв'язання задачі 5.4

Задача 5.5. Створити двовимірний масив A розміром 4×6 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 5).$$

Елементи масиву обчислити за формулою:

$$a_{ij} = \frac{(5j^2 - 4)}{(i - 1, 2j + 2)}.$$

Обчислити суму елементів кожного рядка.

Програмний код:

```
using System;

namespace C139
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 4;        // кількість рядків масиву A
            const int m = 6;        // кількість стовпців масиву A
            int i, j;
            double[,] A = new double[n, m];
            //Обчислення елементів масиву за формулою та виведення їх на екран
            Console.WriteLine("  Масив A  ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                {
                    A[i, j] = (5 * j * j - 4.0) / (i - 1.2 * j + 2);
                    Console.Write($"{A[i,j],9:F2} ");
                }
                Console.WriteLine();
            }
            Console.WriteLine();
            // Обчислення суми елементів
            // кожного рядка матриці
            double s;
            for (i = 0; i < n; i++)
            {
                s = 0;
                for (j = 0; j < m; j++)
                {
```



```

// і виведення їх на екран
Console.WriteLine(" Массив A ");
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        A[i, j] = rnd.NextDouble() * 10;
        Console.Write($"{A[i,j],9:F2} ");
    }
    Console.WriteLine();
}
Console.WriteLine();
//Визначення суми елементів
// кожного стовпця матриці
double s;
for (j = 0; j < m; j++)
{
    s = 0;
    for (i = 0; i < n; i++)
    {
        s = s + A[i, j];
    }
    Console.WriteLine($" Сума елементів стовпця {j} =
                        {s:F3}");
}
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.11.

```

D:\Program\C140\C140\bin\Debug\C140.exe
Массив A
  1,04    1,24    3,07    0,93    4,72
  5,90    4,09    4,42    1,28    1,80
  7,83    4,87    3,49    7,54    2,81
  4,61    4,65    2,13    1,73    7,98

Сума елементів стовпця 0 = 19,387
Сума елементів стовпця 1 = 14,844
Сума елементів стовпця 2 = 13,126
Сума елементів стовпця 3 = 11,467
Сума елементів стовпця 4 = 17,324

```

Рисунок 5.11 – Результати розв’язання задачі 5.6

Задача 5.7. Створити двовимірний масив A розміром 5×5 ,
 $A = (a_{ij}; i = 0,1,\dots,4; j = 0,1,\dots,4)$.

Елементи масиву задати у програмному кодї.

Обчислити суму елементів, розташованих на головній діагоналі матриці.

Програмний код:

```
using System;

namespace C141
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 5; // кількість рядків і стовпців масиву A
            int i, j;
            double[,] A = new double[n, n]
                {{1,2,3,4,5},
                {2,5,8,9,12},
                {7,12,15,18,21},
                {21,22,23,24,25},
                {31,32,35,47,40}};
            //Виведення елементів масиву на екран
            Console.WriteLine(" Масив A ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                {
                    Console.Write($"{A[i,j],9:F2} ");
                }
                Console.WriteLine();
            }
            Console.WriteLine();
            // Обчислення суми елементів, розташованих
            // на головній діагоналі
            double s = 0;
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                {
                    if (i == j)
                    {
```

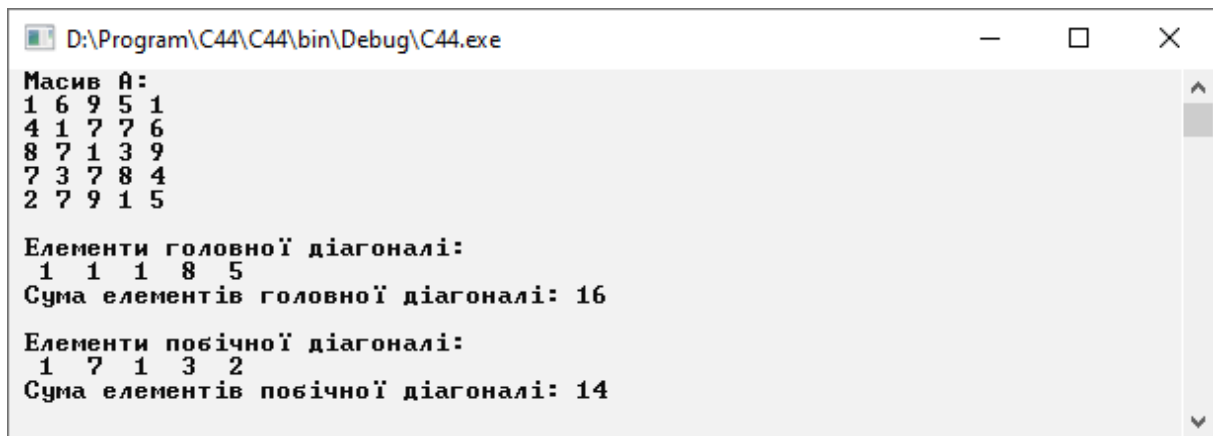


```

// Накопичення та виведення елементів масиву
Console.WriteLine(" Масив A: ");
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        A[i, j] = rnd.Next(1, 10);
        Console.Write(" " + A[i, j]);
    }
    Console.WriteLine();
}
//Створення масивів C[n] та S[n] і накопичення їх сум
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        if (i == j)
        {
            C[i] = A[i, j];
            sumC = sumC + C[i];
        }
        if (i == n - j - 1)
        {
            S[i] = A[i, j];
            sumS = sumS + S[i];
        }
    }
}
Console.WriteLine();
Console.WriteLine(" Елементи головної діагоналі: ");
for (i = 0; i < n; i++) Console.Write(" " + C[i]);
Console.WriteLine($" \n Сума елементів головної
                        діагоналі: {sumC}");
Console.WriteLine(" \n Елементи побічної діагоналі: ");
for (i = 0; i < n; i++) Console.Write(" " + S[i]);
Console.WriteLine($" \n Сума елементів побічної
                        діагоналі: {sumS}");
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.13.



```
D:\Program\C44\C44\bin\Debug\C44.exe
Масив А:
1 6 9 5 1
4 1 7 7 6
8 7 1 3 9
7 3 7 8 4
2 7 9 1 5

Елементи головної діагоналі:
1 1 1 8 5
Сума елементів головної діагоналі: 16

Елементи побічної діагоналі:
1 7 1 3 2
Сума елементів побічної діагоналі: 14
```

Рисунок 5.13 – Результати розв’язання задачі 5.8

Задача 5.9. Створити цілочисловий двовимірний масив *A* розміром 3×4 . Елементи масиву задати у програмному кодї. Визначити середнє арифметичне його елементів і кількість додатних елементів у кожному рядку.

Програмний код:

```
using System;

namespace C40
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 3, m = 4;
            int i, j;
            int[,] A = new int[n, m]
                {{2, -2, 8, 9},
                {-4, -5, 6, -2},
                {7, 0, 1, 1}};
            Console.WriteLine(" Масив А: ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                    Console.Write($"{A[i,j],8} ");
                Console.WriteLine();
            }
            Console.WriteLine();
        }
    }
}
```

```

double sum = 0;    //сума всіх елементів
int k;    //кількість додатних елементів у кожному рядку
double sr;    //середнє арифметичне всіх елементів
for (i = 0; i < n; i++)
{
    k = 0;
    for (j = 0; j < m; j++)
    {
        sum = sum + A[i, j];
        if (A[i, j] > 0)
            k = k + 1;
    }
    Console.WriteLine($" У рядку {i} додатних
                        елементів {k}");
}
sr = sum / (n * m);
Console.WriteLine($" Середнє арифметичне всіх
                    елементів = {sr:F3}");

Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.14.

```

D:\Program\C40\C40\bin\Debug\C40.exe
Массив A:
  2    -2    8    9
 -4   -5    6   -2
  7    0    1    1
У рядку 0 додатних елементів 3
У рядку 1 додатних елементів 1
У рядку 2 додатних елементів 3
Середнє арифметичне всіх елементів = 1,750

```

Рисунок 5.14 – Результати розв’язання задачі 5.9

Пояснення:

```

int[,] A = new int[n, m]
    {{2, -2, 8, 9},
     {-4, -5, 6, -2},
     {7, 0, 1, 1}};

```

Оголошується двовимірний масив A із $n \times m$ цілих чисел типу `int`. Одночасно масив наповнюється потрібними значеннями. У даному випадку, коли задано список ініціалізації, n і m повинні бути тільки константами.

Задача 5.10.

а) Створити двовимірний масив A розміром 4×6 ,
 $A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 5)$.

Масив містить дані типу `double`. Елементи масиву задати у програмному коді.

б) Отримати одновимірний масив $SA(4)$, числові значення елементів дорівнюють середньому арифметичному елементів рядків. Вивести на екран одновимірний масив $SA(4)$. Обчислити добуток елементів одновимірного масиву $SA(4)$.

в) Скоректувати програму для випадку, коли потрібно отримати одновимірний масив, числові значення елементів якого дорівнюють середньому арифметичному значенню елементів стовпців.

Для розв'язання задачі створити консольний додаток.

Програмний код:

```
using System;

namespace C39
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 4;        // кількість рядків масиву A
            const int m = 6;        // кількість стовпців масиву A
            int i, j;
            double s, p;
            double[,] A = new double[n, m]
                {{1,2,3,4,5,6},
                {7,8,9,10,11,12},
                {13,14,15,16,17,18},
                {19,20,21,22,23,24}};

            double[] SA = new double[n];
            // Виведення елементів масиву A на екран
            Console.WriteLine(" Масив A: ");
            for (i = 0; i < n; i++)
            {
```


Задача 5.11. Створити квадратну матрицю A розміром $n \times n$,

$$A = (a_{ij}; i = 0, 1, \dots, (n-1); j = 0, 1, \dots, (n-1)).$$

Дійсні елементи матриці A задаються за допомогою генератора випадкових чисел.

Отримати нову матрицю B розміром $n \times n$. Нова матриця створюється з матриці A , в якій елементи непарних рядків будуть замінені на задане число X . Для розв'язання задачі створити консольний додаток.

Програмний код:

```
using System;

namespace C41
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 5; // кількість рядків і стовпців матриці
            const double x = 7;
            int i, j;
            double[,] A = new double[n, n];
            double[,] B = new double[n, n];
            Random rnd = new Random();
            // Заповнення матриці A
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                    A[i, j] = rnd.NextDouble() * 10 + 7;
            }
            // Виведення елементів початкової матриці A
            Console.WriteLine(" Матриця A: ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                    Console.Write($"{A[i,j],9:F3} ");
                Console.WriteLine();
            }
            Console.WriteLine();
            // Обчислення елементів матриці B
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
```

```

        if (i % 2 == 0)
            B[i, j] = A[i, j];
        else
            B[i, j] = x;
    }
    Console.WriteLine();
    // Виведення нової матриці B
    Console.WriteLine(" Матриця B: ");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
            Console.Write($"{B[i,j],9:F3} ");
        Console.WriteLine();
    }
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 5.16.

Матриця А:				
12,955	10,844	12,221	8,240	15,160
12,998	15,407	14,893	7,078	7,352
8,369	11,764	15,603	12,229	10,993
13,752	15,993	15,912	15,574	12,655
11,590	13,596	14,103	10,379	7,381

Матриця В:				
12,955	10,844	12,221	8,240	15,160
7,000	7,000	7,000	7,000	7,000
8,369	11,764	15,603	12,229	10,993
7,000	7,000	7,000	7,000	7,000
11,590	13,596	14,103	10,379	7,381

Рисунок 5.16 – Результати розв’язання задачі 5.11

Пояснення:

```

if (i % 2 == 0)
    B[i, j] = A[i, j];
else
    B[i, j] = x;

```

Якщо остача від ділення номера рядка дорівнює нулю (тобто рядок парний), то елементу матриці В присвоюється значення елемента матриці А, інакше елементу матриці В присвоюється значення x.

5.2. Завдання для виконання лабораторних робіт

Задача 1. Створити двовимірний масив A розміром 4×6 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 5).$$

Елементи масиву задати за допомогою генератора випадкових чисел.

Обчислити добуток елементів кожного рядка.

Задача 2. Створити двовимірний масив A розміром 4×5 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 4).$$

Елементи масиву задати за допомогою генератора випадкових чисел.

Обчислити добуток елементів кожного стовпця.

Задача 3. Створити двовимірний масив A розміром 5×5 ,

$$A = (a_{ij}; i = 0, 1, \dots, 4; j = 0, 1, \dots, 4).$$

Елементи масиву задати за допомогою генератора випадкових чисел.

Обчислити суму елементів, розташованих над головною діагоналлю.

Задача 4. Створити двовимірний масив A розміром 6×6 ,

$$A = (a_{ij}; i = 0, 1, \dots, 5; j = 0, 1, \dots, 5).$$

Елементи масиву задати за допомогою генератора випадкових чисел.

Обчислити суму елементів, розташованих під головною діагоналлю.

Задача 5. Створити двовимірний масив A розміром 5×5 ,

$$A = (a_{ij}; i = 0, 1, \dots, 4; j = 0, 1, \dots, 4).$$

Елементи масиву задати в програмному коді.

$$A = \begin{pmatrix} 1 & 3 & 5 & 8 & 10 \\ -5 & 7 & -9 & 12 & 15 \\ 16 & -18 & 20 & -23 & 24.5 \\ -15.8 & 22 & 24 & -25 & 30 \\ 32 & 35.8 & -41 & 45 & 50 \end{pmatrix}.$$

Сформувати одновимірний масив із елементів головної діагоналі.

Задача 6. Створити двовимірний масив A розміром 4×6 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 5).$$

Елементи масиву розрахувати за формулою:

$$a_{ij} = \frac{(5j^2 - 4)}{(i - 1, 2j + 2)}.$$

Обчислити кількість додатних (або від'ємних) елементів масиву в кожному рядку.

Задача 7. Створити двовимірний масив A розміром 4×6 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 5).$$

Елементи масиву розрахувати за формулою:

$$a_{ij} = \frac{(5j^2 - 4)}{(i - 1, 2j + 2)}.$$

Обчислити кількість додатних (або від'ємних) елементів масиву у кожному стовпці.

Задача 8. Створити двовимірний масив A розміром 4×5 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 4).$$

Елементи масиву задати в програмному кодї.

$$A = \begin{pmatrix} 3 & 5 & 11 & 8 & 7 \\ 42 & 10 & 45 & 31 & 14 \\ 5 & 2 & 12 & 3 & 14 \\ 80 & 18 & 22 & 35 & 40 \end{pmatrix}.$$

Визначити суму (або добуток) елементів за периметром матриці.

Задача 9. Створити двовимірний масив A розміром 4×7 ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 6).$$

Елементи масиву розрахувати за формулою:

$$a_{ij} = 10^i - 2^j.$$

Знайти максимальний елемент масиву та його індекси.

Задача 10. Створити двовимірний масив A розміром 8×5 ,

$$A = (a_{ij}; i = 0, 1, \dots, 7; j = 0, 1, \dots, 4).$$

Елементи масиву розрахувати за формулою:

$$a_{ij} = 20i - (0,5j)^2.$$

Знайти мінімальний елемент масиву та його індекси.

Задача 11. Створити двовимірний масив A розміром 5×8 ,

$$A = (a_{ij}; i = 0, 1, \dots, 4; j = 0, 1, \dots, 7).$$

Елементи масиву розрахувати за формулою:

$$a_{ij} = i^2 - 5j^2 + 30.$$

Отримати транспоновану матрицю і надрукувати її за рядками. Для транспонування матриці необхідно замінити рядки матриці її стовпцями, а стовпці – рядками, тобто обчислити

$$b_{ij} = a_{ji}; i = 0, 1, \dots, 7; j = 0, 1, \dots, 4.$$

Задача 12. Створити двовимірний масив A розміром 3×5 ,

$$A = (a_{ij}; i = 0, 1, \dots, 2; j = 0, 1, \dots, 4).$$

Елементи масиву задати у програмному коді

$$A = \begin{pmatrix} 2 & 7,5 & -12 & 45 & -34 \\ 70 & 85,5 & -56 & 92 & 102 \\ 115 & -78 & 74 & -100 & 80 \end{pmatrix}.$$

Сформувати одновимірний масив із додатних його елементів, значення елементів піднести до квадрата.

СПИСОК ЛІТЕРАТУРИ

1. Соловей Л.В. Основи програмування мовою С#: навч. посібник / Л. В. Соловей, Н. М. Мірошніченко, Т. Г. Бабак, О. О. Голубкіна, Є. Д. Пономаренко. – Харків: НТУ «ХП», 2019. – 488 с.

2. Коноваленко І. В. Платформа .NET та мова програмування С# 8.0 : Навчальний посібник/ І. В. Коноваленко, П. О. Марущак. – Тернопіль: ФОП Паляниця В. А., 2020. – 320 с.

3. Методичні вказівки для виконання розрахункових завдань «Програмування мовою С#» з дисциплін «Інформаційні технології та програмування», «Обчислювальна математика і програмування» для студентів хімічних спеціальностей усіх форм навчання/ Укл. Л. В. Соловей, Н. М. Мірошніченко, Є. Д. Пономаренко. – Харків: НТУ «ХП», 2021. – 64 с.

Навчальне видання

СОЛОВЕЙ Людмила Валентинівна
МІРОШНІЧЕНКО Наталія Миколаївна
БАБАК Тетяна Геннадіївна

ПРОГРАМУВАННЯ МОВОЮ C# Visual Studio 2019

Лабораторний практикум

для студентів хімічних спеціальностей усіх форм навчання

У трьох частинах

Частина 1

Відповідальний за випуск проф. Ведь В.Є.

Роботу до видання рекомендував проф. Рисований О.М.

В авторській редакції

План 2021 р., поз. 37

Підписано до друку 22.02.22 р. Формат 60×84 1/16. Папір офсетний.
Riso-друк. Гарнітура Таймс. Ум. друк. арк. Наклад 50 прим.
Зам № Ціна договірна

Видавничий центр НТУ «ХП», вул. Кирпичова, 2, м. Харків-2, 61002
Свідоцтво суб'єкта видавничої справи ДК №5478 від 21.08.2017 р.
