

ЗМІСТ

Рішення задач у вигляді Windows-додатків

Тема 1. СКЛАДАННЯ ПРОГРАМ ЛІНІЙНОЇ СТРУКТУРИ	4
Тема 2. СКЛАДАННЯ ПРОГРАМ РОЗГАЛУЖЕНОЇ СТРУКТУРИ	21
Тема 3. ЦИКЛИ	25
Тема 4. ОДНОВИМІРНІ МАСИВИ	28
Тема 5. ДВОВИМІРНІ МАСИВИ	54
Тема 6. МЕТОДИ	67
Тема 7. РЯДКИ	79
Тема 8. ФАЙЛИ	84
Тема 9. ГРАФІКИ ФУНКЦІЙ	87

Тема 1. СКЛАДАННЯ ПРОГРАМ ЛІНІЙНОЇ СТРУКТУРИ

Мета роботи: навчитись розробляти Windows-додатки, вивчити елементи управління: мітка Label, кнопка Button, поле введення TextBox.

Задача 1.1. 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 $a = 3,2$; $b = 17,5$; $x = -4,8$.

Створити *Windows-додаток*. Значення початкових даних увести за допомогою текстових полів, результати розрахунку вивести на мітку. Блок-схему розв'язання задачі подано на рис. 1.1.

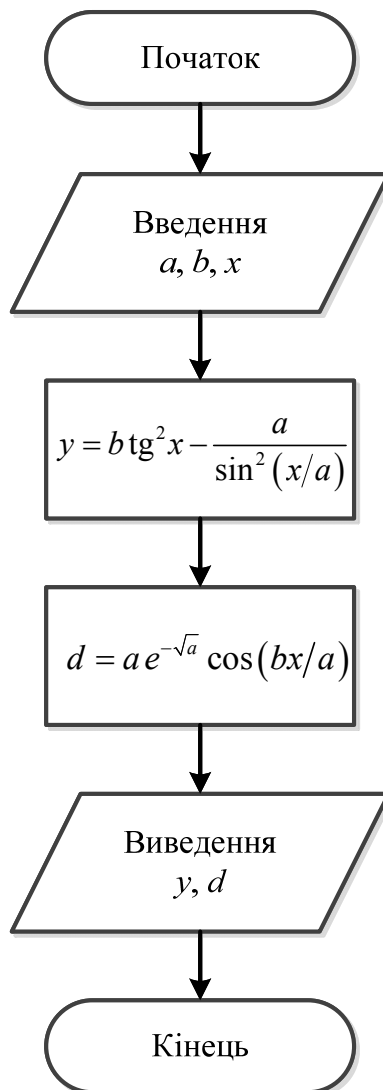


Рисунок 1.1 – Блок-схема алгоритму до задачі 1.1

Для створення *Windows-додатка* необхідно:

1. Завантажити систему Microsoft Visual Studio 2019

Після завантаження Visual Studio відображується вікно, основний простір якого займає початкова сторінка *Microsoft Visual Studio*, яка містить команди створення й відкриття проектів, посилання на останні проекти, довідкові матеріали, розширення Visual Studio, ресурси співтовариства. Початковий інтерфейс користувача наведено на рис. 1.2.

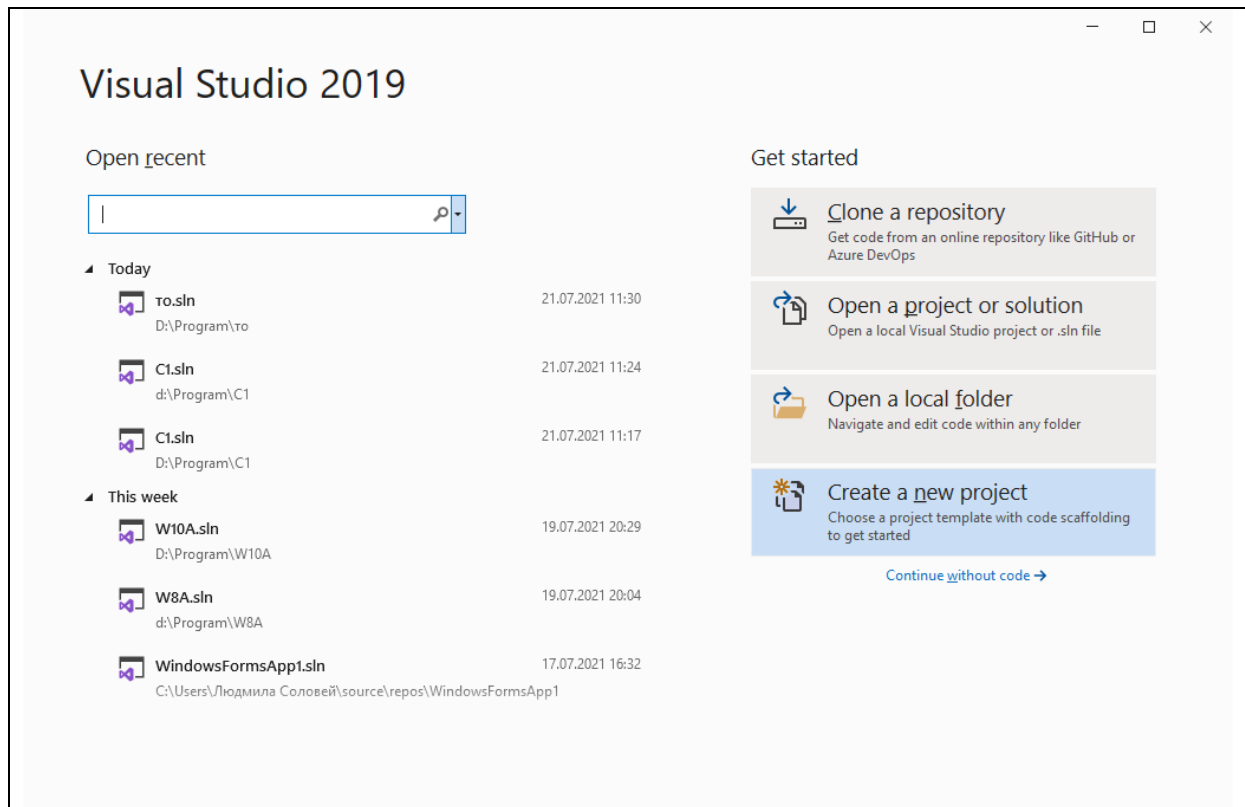


Рисунок 1.2 – Початковий інтерфейс користувача Visual C# 2019

2. Створити проект

Викликати команду *Create a new project* (Створити проект).

У вікні *Create a new project* (рис. 1.3) зверху є кілька випадаючих списків:

All languages (Усі мови) – C#.

All platforms (Всі платформи) – Windows

All Project type (Всі типи проектів) – Desktop (робочий стіл)

У середньому стовпчику виберемо шаблон (Templates) *Приложение Windows Forms App (.NET Framework)* (рис. 1.4)

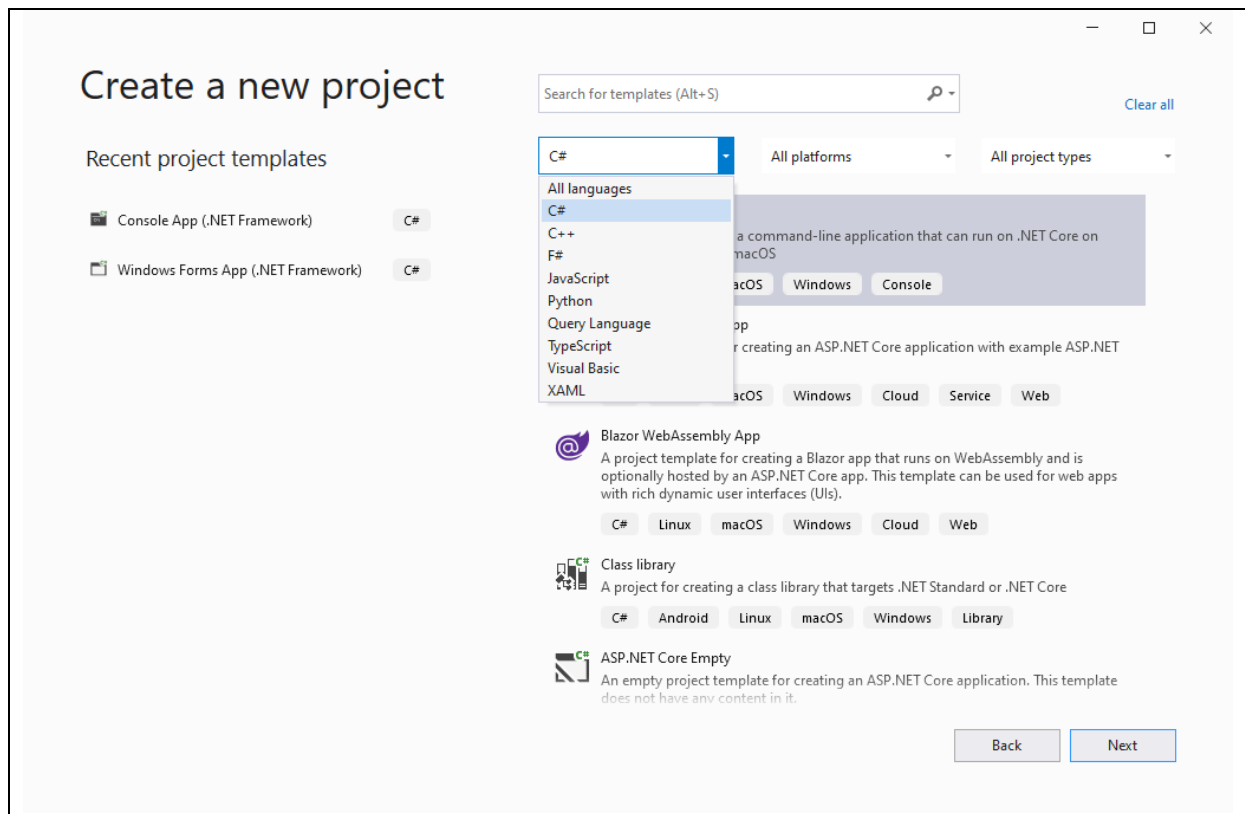


Рисунок 1.3— Вікно *Create a new project*

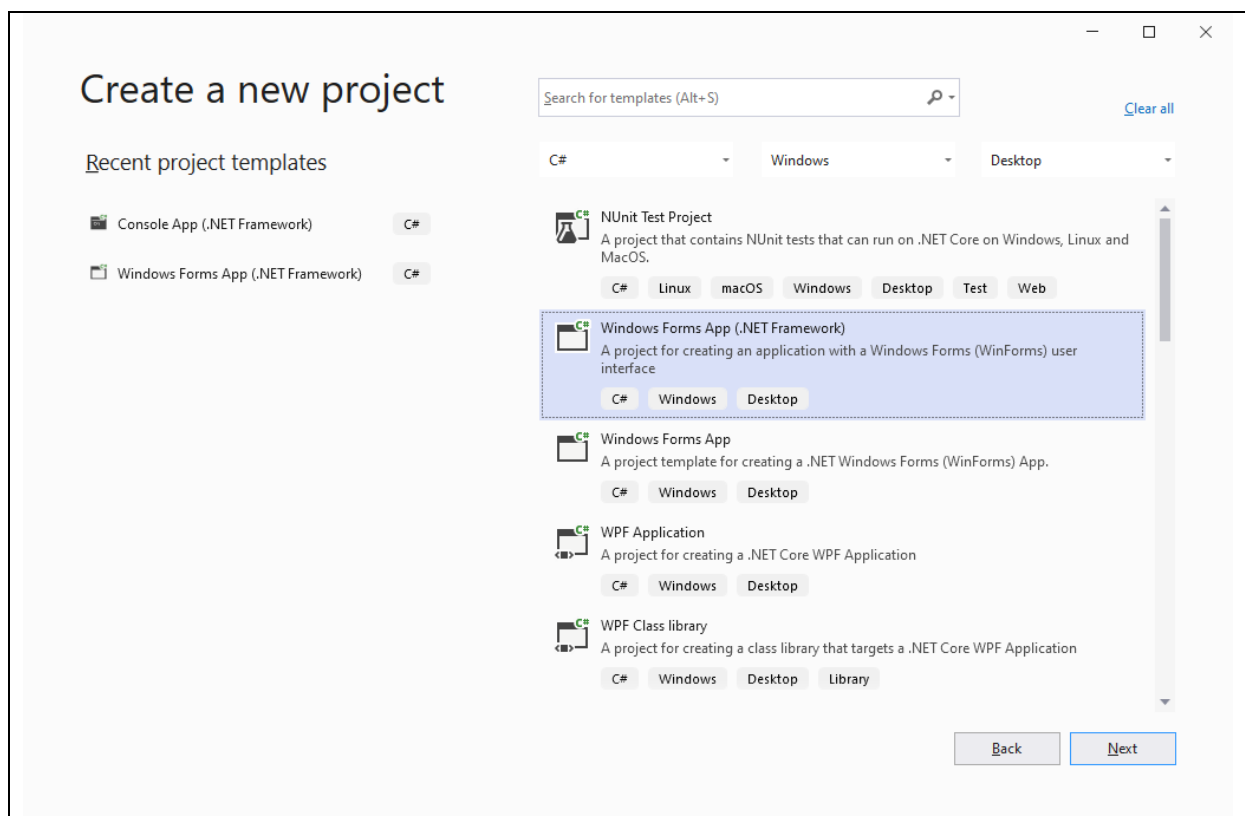


Рисунок 1.4— Вікно *Create a new project*

Після створення windows-додатка в новому вікні *Configure your new project* (рис. 1.5) задаємо:

- у полі *Project name* введіть ім'я проекту – W1 (за замовчуванням система задає ім'я *WinFormApp1*)
- 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.

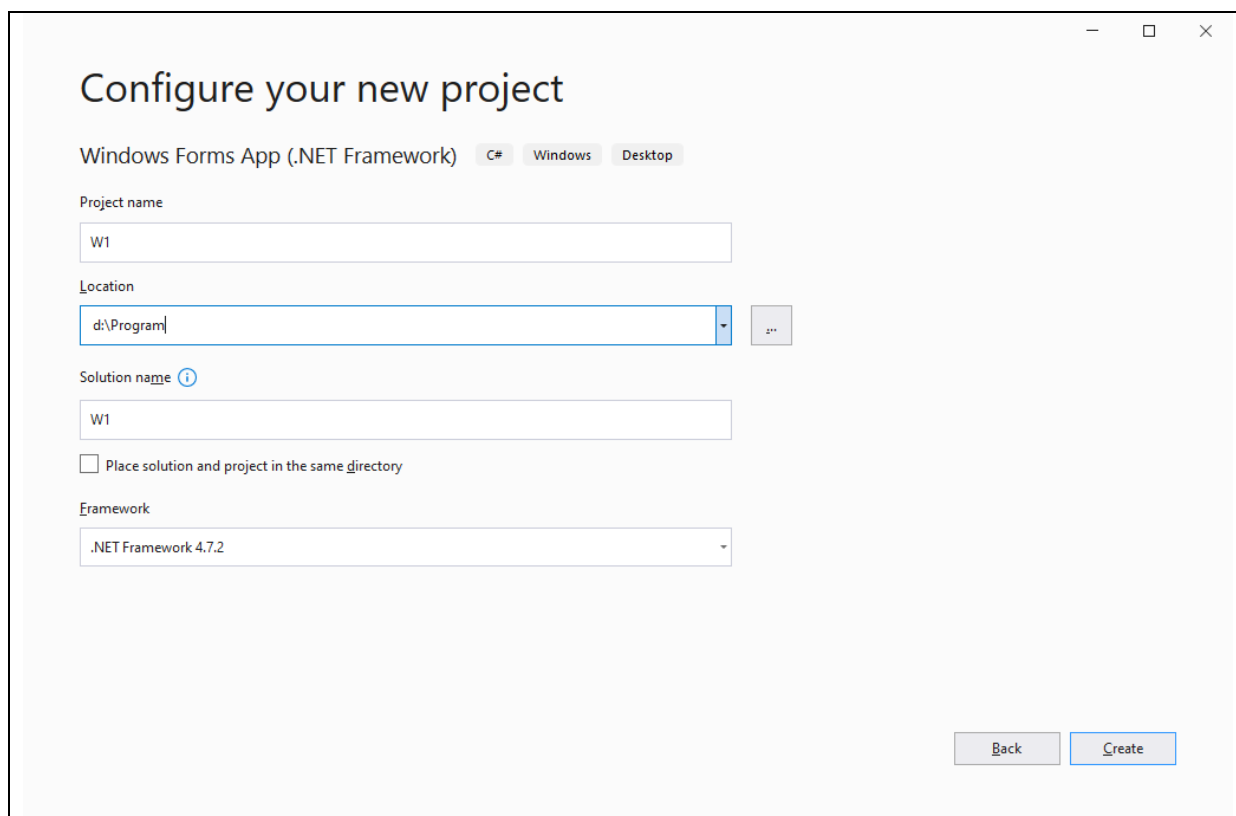


Рисунок 1.5 – Вікно *Configure your new project*

При створенні нового проекту середовище сформує шаблон Windows-додатка. У результаті побачимо вікно, наведене на рис. 1.6. У цьому вікні зображена екранна форма – **Form1**, у якій програмісти розташовують різні компоненти графічного інтерфейсу користувача або, як їх інакше називають, елементи керування. Це поля для введення тексту **Textbox**, командні кнопки **Button**, мітки **Label** та ін.

3. *Перевірити, щоб були завантажені панель елементів і панель властивостей*

View/Toolsbox/All windows forms

(Вид / Панель елементів / Стандартні елементи управління)

View/Properties window

(Вид / Вікно властивостей)

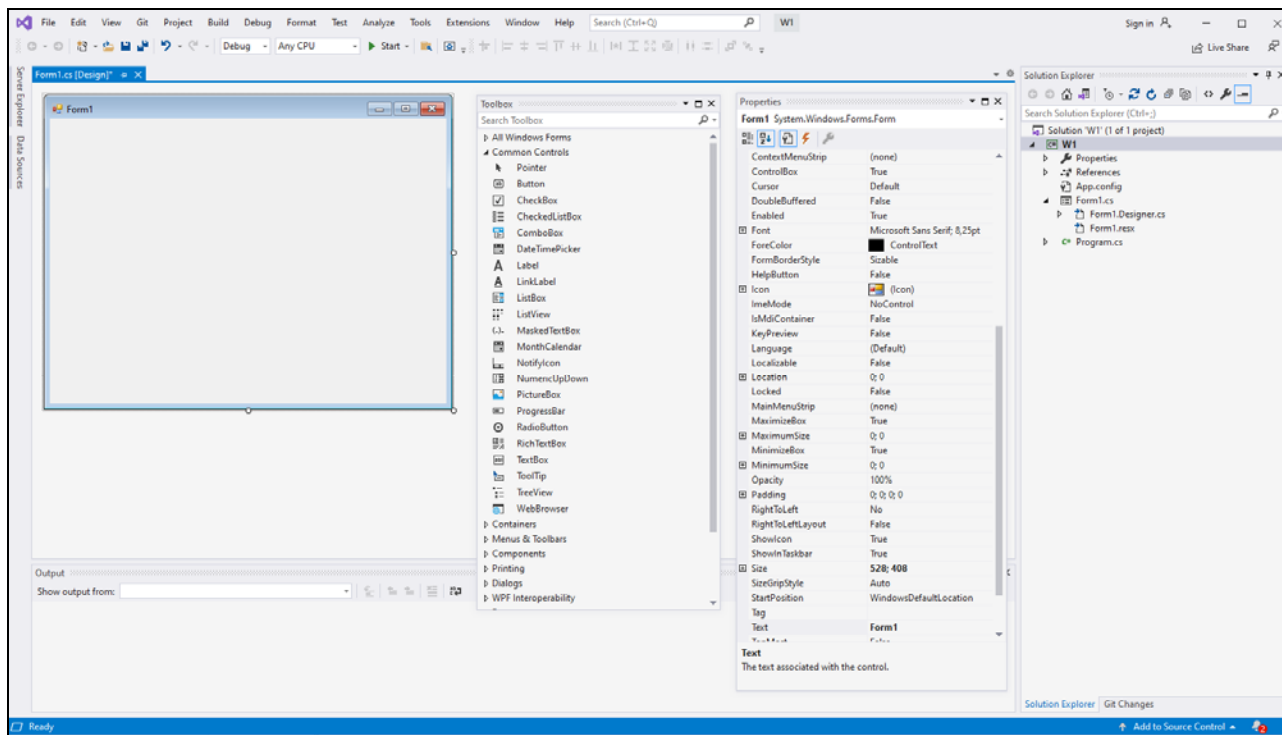


Рисунок 1.6 – Вікно для проектування інтерфейсу користувача

4. На формі *Form1* створіть об'єкти, як показано на рис. 1.7, 1.8.

На формі розташуйте:

- чотири мітки (Label),
- три текстових поля (TextBox),
- дві командні кнопки (Button).

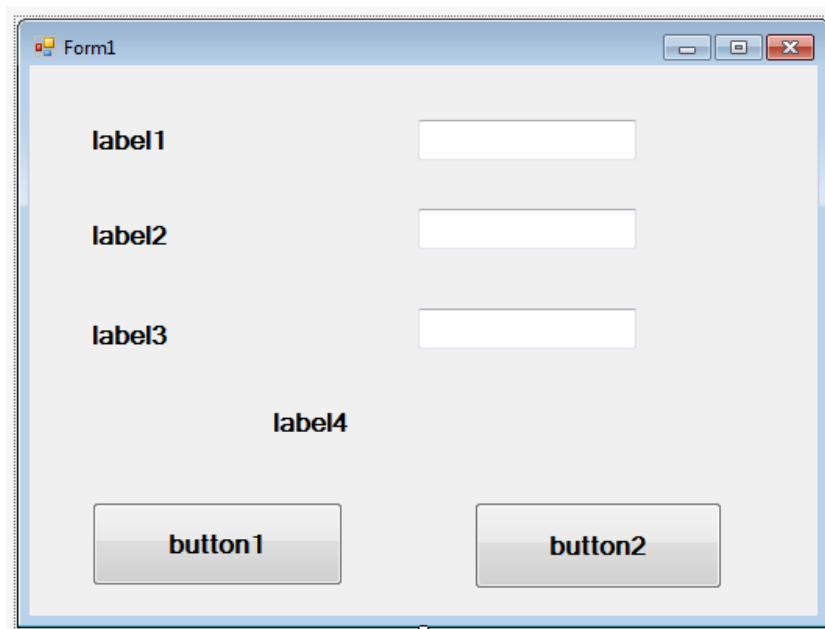


Рисунок 1.7 – Форма до задачі 1.1

5. Установіть значення властивостей об'єктів відповідно до табл. 1.1.

Таблиця 1.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1	Text	Тема 1 Програма лінійної структури
Мітка	Label1	Text	Введіть значення а:
Мітка	Label2	Text	Введіть значення b:
Мітка	Label3	Text	Введіть значення x:
Мітка	Label4		
Виділіть мітки Label1-Label4 (клавіша[Shift] +миша) і установіть однакові для всіх міток властивості		TextAlign	TopLeft
		Font	Times New Roman, Regular, 12
Текстове поле	TextBox1	Name	txta
		Text	3,2
Текстове поле	TextBox2	Name	txtb
		Text	17,5
Текстове поле	TextBox3	Name	txtx
		Text	-4,8
Виділіть текстові поля TextBox1 – TextBox3 і установіть однакові для всіх полів властивості		TextAlign	Left
		Font	Times New Roman, Regular, 12
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
		Font	Times New Roman, Bold, 12
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи
		Font	Times New Roman, Bold, 12

Рисунок 1.8 – Форма до задачі 1.1

Угода про імена об'єктів

Іноді для підвищення інформативності об'єктів (елементів керування), що застосовані у проекті, програмісти використовують в іменах префікси, які визначають їхню належність до певного класу. Формат імені об'єкта:

ПрефіксІм'я,

де Ім'я – інформативне з точки зору функціонального призначення ім'я об'єкта.

Нижче наведено приклади використання префіксів для задання імен об'єктів.

Префікс	Елемент керування	Приклад
txt	TextBox (текстове поле)	txtA
cmd	Button (Кнопка керування)	cmdStart
pic	PictureBox (Рисунок)	picRis

6. Програмування

Перш ніж розпочати програмування, необхідно визначити ті події, для яких необхідно розробити алгоритми й описати їх мовою програмування. У нашому прикладі є такі події: *клацання мишею по командній кнопці <Обчислити> і клацання мишею по командній кнопці <Завершення роботи>*. Створимо методи для обробки цих подій (оброблювачі подій). У тілі оброблювачів подій програміст пише код, який буде виконуватися при настанні події.

Додавання оброблювачів подій

Двічі клацніть на кнопці <Обчислити>. Це приведе безпосередньо до оброблювача події елемента керування, що використовується за замовчуванням, – для кнопки такою подією є подія **Click**, тому саме його оброблювач створюється. Після подвійного клацання на кнопці <Обчислити> відкриється *Вікно програмного коду*. У вікні редактора з'явиться створений автоматично шаблон оброблювача події:

```
private void cmdStart_Click(object sender, EventArgs e)
{

}
```

Як ім'я методу присвоюється ім'я елемента керування, за яким іде символ підкреслення й ім'я оброблюваної події – **cmdStart_Click** (рис. 1.9). Заповнимо цей оброблювач.

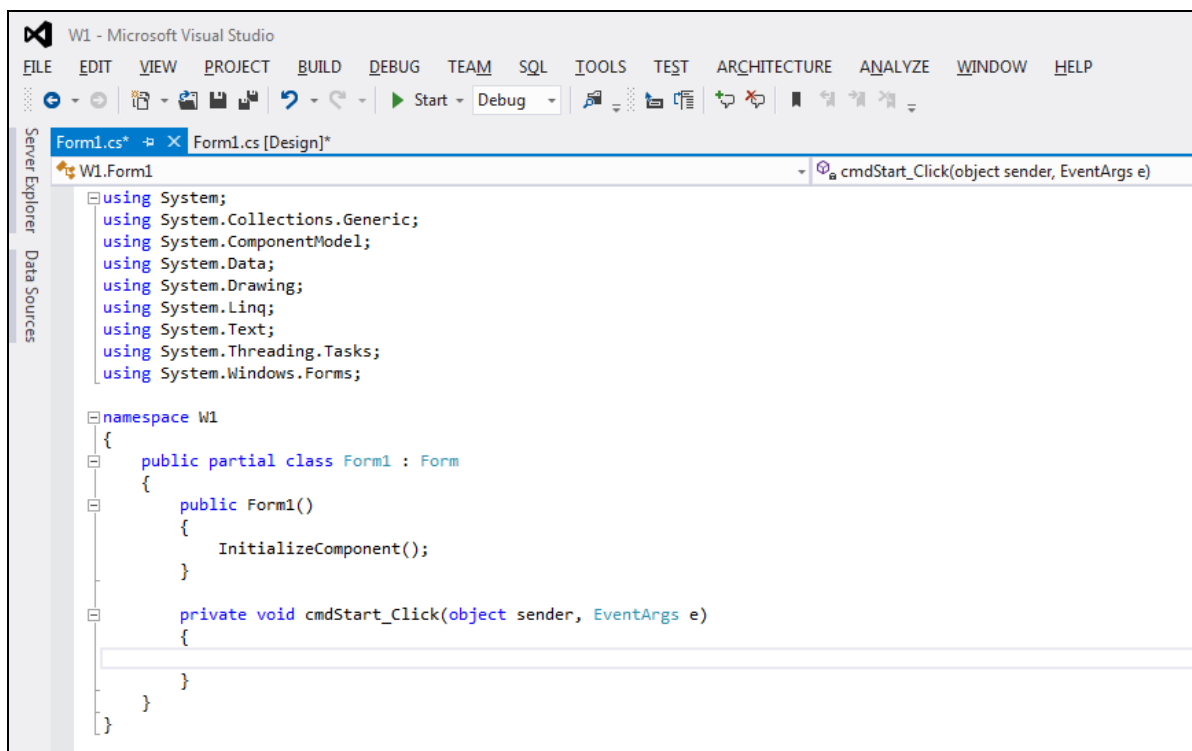


Рисунок 1.9 – Вкладка програмного коду

Для написання методу **cmdEnd_Click** обробки події клацання мишею по командній кнопці <Завершення роботи> двічі клацніть по кнопці <Завершення роботи>. Відкриється *Вікно програмного коду* зі створеним автоматично шаблоном оброблювача події. Заповнимо цей оброблювач.

Програмний код:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace W1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void cmdStart_Click(object sender, EventArgs e)
            {
                double a, b, x, y, d;
                // Зчитування значення a
                a = Convert.ToDouble(txta.Text);
                // Зчитування значення b
                b = Convert.ToDouble(txtb.Text);
                // Зчитування значення x
                x = Convert.ToDouble(txtx.Text);
                // Обчислюємо арифметичний вираз
                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);
                // Виводимо результати розрахунку на мітку
                label4.Text = $"Результати розрахунку: \n y= {y:F3}
                    d= {d:F4} ";
            }

            private void cmdEnd_Click(object sender, EventArgs e)
            {
                // Завершення роботи додатка
                Close();
            }
        }
    }
}
```

7. 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.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace W1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void cmdStart_Click(object sender, EventArgs e)
            {
                double a, b, x, y, d;
                // Зчитування значення a
                a = Convert.ToDouble(txta.Text);
                // Зчитування значення b
                b = Convert.ToDouble(txtb.Text);
                // Зчитування значення x
                x = Convert.ToDouble(txtx.Text);
                // Обчислюємо арифметичний вираз
                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);
                // Виводимо результати розрахунку на мітку
                label4.Text = $"Результати розрахунку: \n y= {y:F3}
                    d= {d:F4} ";
            }

            private void cmdEnd_Click(object sender, EventArgs e)
            {
                // Завершення роботи додатка
                Close();
            }
        }
    }
}
```

8. Запустіть програму на виконання

Debug/ Start Debugging (Отладка/ Почати отладку)
або натисніть на зелену стрілку ► *Start* на панелі інструментів

При цьому відбувається трансляція й, якщо немає помилок, компонування програми й створення єдиного файлу, що завантажується, з розширенням .exe. На екрані з'являється активне вікно програми.

Клацніть на кнопці <Обчислити> лівою кнопкою миші. Результати розрахунків подано на рис. 1.10.

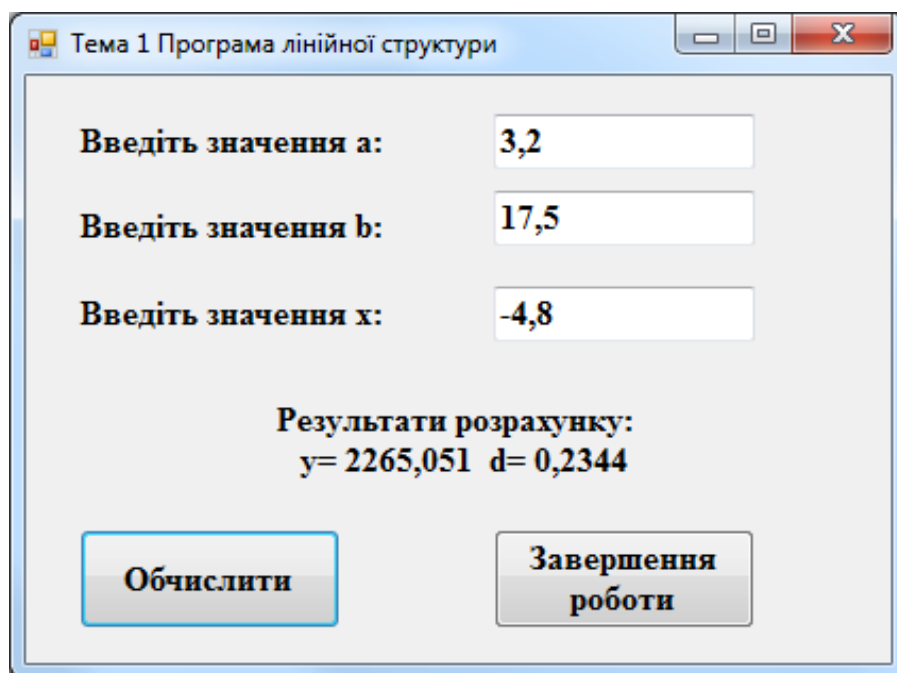


Рисунок 1.10 – Результати розв'язання задачі 1.1

9. Клацніть на кнопці <Завершення роботи> лівою кнопкою миші

10. Збережіть проект

File/ Save All (Файл / Зберегти все)

Проект у Visual Studio складається з файлу проекту (файл із розширенням .csproj), одного або декількох файлів початкового тексту (з розширенням .cs), файлів з описом вікон форми (з розширенням .designer.cs), файлів ресурсів (з розширенням .resx), а також ряду службових файлів. У файлі проекту знаходиться

інформація про модулі, що становлять цей проект, ресурси, що входять у нього, а також параметри побудови програми.

Файл проекту автоматично створюється та змінюється середовищем Visual Studio і не призначений для ручного редагування. Файл початкового тексту – програмний модуль, призначений для розміщення текстів програм. У цьому файлі програміст розміщує текст програми, написаний мовою C#.

При компіляції програми Visual Studio створює файли, що виконуються, `.exe` – файли в каталозі `bin`.

Відкомпільована програма знаходиться в папці

`D:\ProgramVS\C1\C1\bin\Debug\C1.exe`

Пояснення до програми:

1) `using System.Windows.Forms;`

Простір `System.Windows.Forms` містить величезну кількість типів, що є будівельними блоками Windows-додатків.

2) `a = Convert.ToDouble(txta.Text);`

Оператор присвоювання. Змінній `a` привласнити значення, що вводиться в текстове поле з іменем `txta`. Visual C# вважає вміст текстового поля текстом, а не числом, якщо навіть він складається із цифр. Отже, змінні, що беруть своє значення з текстового поля, мова Visual C# теж вважає текстовими, а не числовими. Функція `Convert.ToDouble` конвертує (перетворює) текст у числове дійсне значення подвійної точності (конвертує рядкову змінну `txta.Text` в число `a`)

3) `label4.Text = $"Результати розрахунку: \n y= {y:F3} d= {d:F4} ";`

Виведення даних у поле `label4`. Для виведення на мітку використовується властивість `Text` – напис на мітці. При виведенні числа необхідно перетворити його в послідовність символів, тобто в рядок.

В даному прикладі знак `$` перед рядком вказує, що буде здійснюватися інтерполяція рядків. 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.

`\n` – керуюча послідовність – переведення рядка.

Задача 1.2. Відомі щільність ρ , висота h й діаметр основи D циліндричного злитка, отриманого в металургійній лабораторії (рис. 1.11). Знайти об'єм V , масу m й площу основи злитка S .

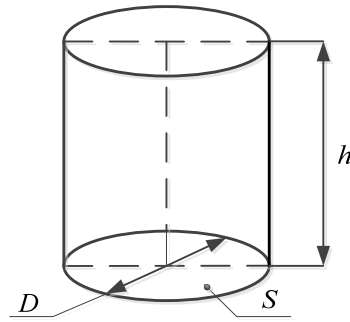


Рисунок 1.11 – Циліндричний злиток до задачі 1.2

Вхідні дані: $\rho = 8940 \frac{\text{кг}}{\text{м}^3}$, $h = 0,15 \text{ м}$, $D = 0,05 \text{ м}$.

Вихідні дані: S , V , m .

Розрахункові формули: $S = \frac{\pi D^2}{4}$; $V = S \cdot h$; $m = \rho \cdot V$.

Створити *Windows-додаток*. Значення вхідних даних увести за допомогою текстових полів, результати розрахунку вивести також у текстове поле. Блок-схему розв'язку задачі подано на рис. 1.12.

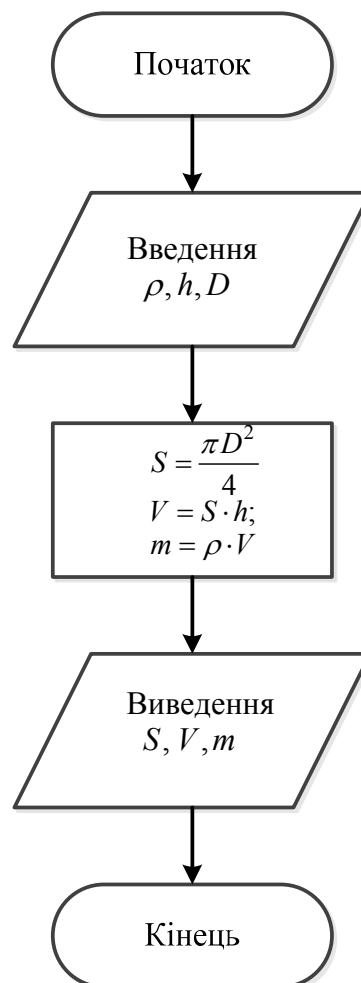


Рисунок 1.12 – Блок-схема алгоритму до задачі 1.2

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 1.13, 1.14.

На формі розташувати:

- п'ять міток (Label),
- чотири текстових поля (TextBox),
- дві командні кнопки (Button).

Рисунок 1.13 – Форма до задачі 1.2

2. Установіть значення властивостей об'єктів відповідно до табл. 1.2.

Таблиця 1.2 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1	Text	Розрахунок маси циліндричного злитка
Мітка	Label1	Text	Вхідні дані
Мітка	Label2	Text	Діаметр злитка, D [м]
Мітка	Label3	Text	Висота, h [м]
Мітка	Label4	Text	Щільність, ρ [кг/м ³]
Мітка	Label5	Text	Результати розрахунку

Закінчення табл. 1.2

1	2	3	4
Виділити мітки Label1-Label5 (клавіша[Shift] +миша) й установити однакові для всіх міток властивості		TextAlign	TopLeft
		Font	Times New Roman, Regular, 12
Текстове поле	TextBox1	Name	txtD
		Text	0,05
Текстове поле	TextBox2	Name	txth
		Text	0,15
Текстове поле	TextBox3	Name	txtro
		Text	8940
Текстове поле	TextBox4	Name	txtRez
		Multiline	True (текстове поле можна розтягувати не тільки по горизонталі, але й по вертикалі)
		ScrollBars	Both (у вікні з'явиться вертикальна, а при необхідності й горизонтальна прокрутки)
Виділити текстові поля TextBox1 – TextBox4 й установити однакові для всіх полів властивості		TextAlign	Left
		Font	Times New Roman, Regular, 12
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
		Font	Times New Roman, Bold, 12
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи
		Font	Times New Roman, Bold, 12

Розрахунок маси циліндричного злитка

Вхідні дані

Діаметр злитка, D [м] 0,05

Висота, h [м] 0,15

Щільність, ρ [кг/м³] 8940

Результати розрахунку

Обчислити

Завершення роботи

Рисунок 1.14 – Форма до задачі 1.2

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA24
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void cmdStart_Click(object sender, EventArgs e)
            {
                double D, h, ro, S, V, m;
                D = Convert.ToDouble(txtD.Text);
                h = Convert.ToDouble(txth.Text);
                ro = Convert.ToDouble(txtro.Text);
                S = Math.PI * D * D / 4.0;
                V = S * h;
                m = ro * V;
                txtRez.Text = txtRez.Text + $"Об'єм V= {V:F5} м³ " +
                    Environment.NewLine +
                    $"Маса m= {m:F5} кг " + Environment.NewLine +
                    $"Площа основи злитка S= {S:F5} м² ";
            }

            private void cmdEnd_Click(object sender, EventArgs e)
            {
                //Процедура закриття форми
                Close();
            }
        }
    }
}
```

Результати розрахунку подано на рис. 1.15.

Розрахунок маси циліндричного злитка

Вхідні дані

Діаметр злитка, D [м]

Висота, h [м]

Щільність, го [кг/м³]

Результати розрахунку

Об'єм V= 0,00029 м³
Маса m= 2,63305 кг
Площа основи злитка S= 0,00196 м²

Обчислити

Завершення роботи

Рисунок 1.15 – Результати розв'язання задачі 1.2

Пояснення до програми:

- 1) `txtRez.Text = txtRez.Text + $"Об'єм V= {V:F5} м³ " + Environment.NewLine + $"Маса m= {m:F5} кг " + Environment.NewLine + $"Площа основи злитка S= {S:F5} м² ";`

Виведення результатів розрахунків у текстове поле `txtRez`. Для виведення в текстове поле використовується властивість `Text`.

При виведенні числа необхідно перетворити його в послідовність символів, тобто в рядок.

`Environment.NewLine` – переведення курсора на новий рядок.

Тема 2. СКЛАДАННЯ ПРОГРАМ РОЗГАЛУЖЕНОЇ СТРУКТУРИ

Мета роботи: вивчити елементи керування – мітку Label, кнопку Button, поле введення TextBox, список ListBox.

Задача 2.1. 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

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

Створимо *Windows-додаток*. Значення a і b задано в програмному коді, значення y вводимо в текстове поле TextBox, результати розрахунку виводимо в поле списку ListBox.

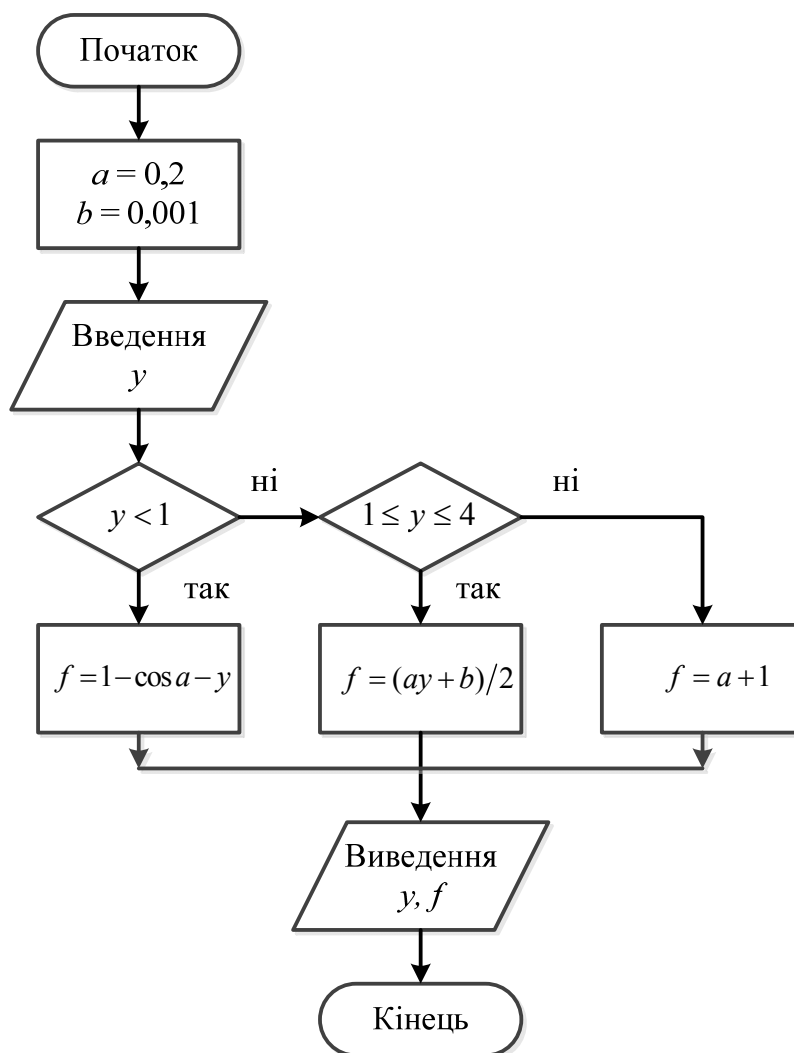


Рисунок 2.1 – Блок-схема алгоритму до задачі 2.1

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 2.2, 2.3.

На формі розташувати:

- дві мітки (Label);
- одне текстове поле (TextBox);
- одне поле списку (ListBox);
- дві командні кнопки (Button).

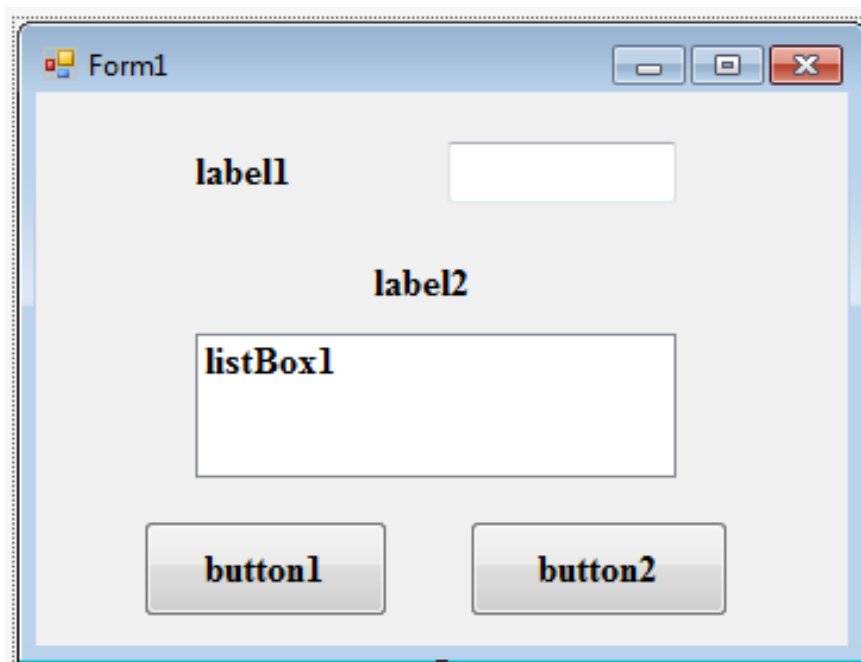


Рисунок 2.2 – Форма до задачі 2.1

2. Установіть значення властивостей об'єктів згідно з табл. 2.1.

Таблиця 2.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Мітка	Label1	Text	Введіть у:
Мітка	Label2	Text	Результати розрахунку
Текстове поле	TextBox1	Name	txty
Поле списку	ListBox1		
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
		Font	Times New Roman, Bold, 12
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи
		Font	Times New Roman, Bold, 12

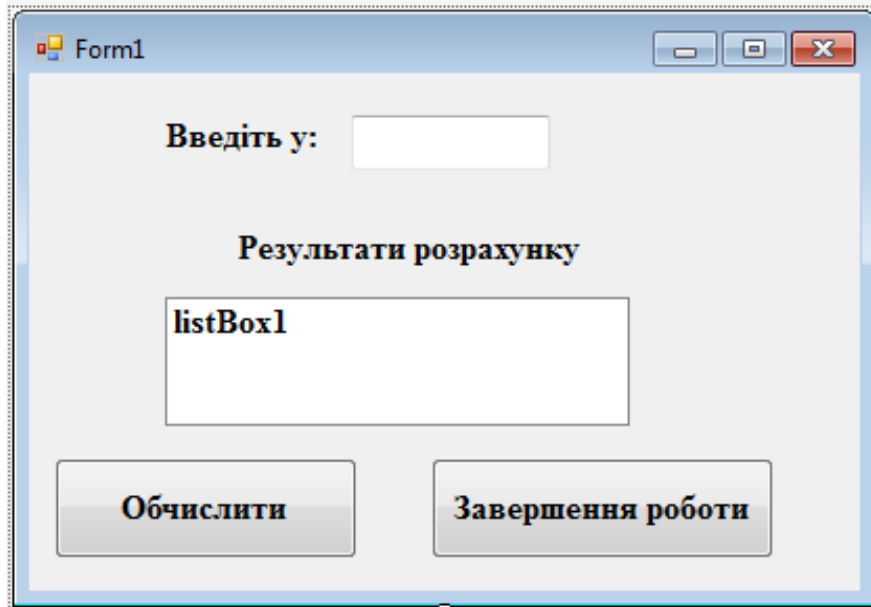


Рисунок 2.3– Форма до задачі 2.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            double a, b, y, f;
            string str;
            a = 0.2; b = 0.001;
            // Обчислення виразу
            y = Convert.ToDouble(txtY.Text);
            if (y < 1)
                f = 1 - Math.Cos(a) - y;
            else if (1 <= y && y <= 4)
```

```

        f = (a * y + b) / 2;
    else
        f = a + 1;
    str = $"y = {y,7:F2};    f = {f,7:F3}";
    // Виведення результатів розрахунку в поле ListBox
    listBox1.Items.Add(str);
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 2.4.

Рисунок 2.4 – Результати розв’язання задачі 2.1

Пояснення:

1)string str;

Оголошується рядкова змінна

2)str = \$"y = {y,7:F2}; f = {f,7:F3}";

Формується рядок, що містить відформатовані дані.

3)listBox1.Items.Add(str);

Виведення даних (текстового рядка str) у поле ListBox. Елемент керування ListBox – поле списку. Для додавання елементів у список програмним способом призначений метод Add колекції Items елемента ListBox.

Тема 3. ЦИКЛИ

Мета роботи: вивчити елементи керування — мітку `Label`, кнопку `Button`, список `ListBox`

Задача 3.1. Написати програму, яка виводить таблицю значень функції в діапазоні від -2 до 2 , з кроком $0,5$. Таблиця повинна складатися з двох стовпців: значень аргументу і відповідних їм значень функції. Для розв’язання задачі використовувати оператори циклів: `for`, `while`, `do-while`.

Створити Windows-додаток із використанням циклу `for`.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об’єкти, як показано на рис. 3.1, 3.2.

На формі розташувати:

- одну мітку (`Label`);
- одне поле списку (`ListBox`);
- дві командні кнопки (`Button`).

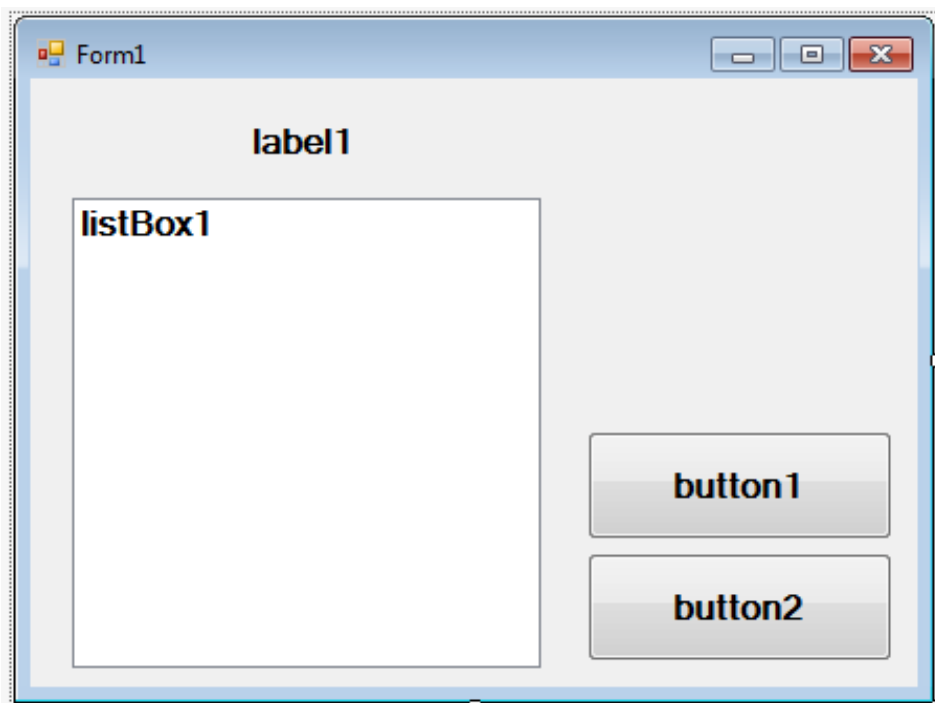


Рисунок 3.1 – Форма до задачі 3.1

2. Установіть значення властивостей об’єктів згідно з табл. 3.1.

Таблиця 3.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Мітка	Label	Text	Результати розрахунку
Поле списку	ListBox1	ScrollAlwaysVisible	True
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи

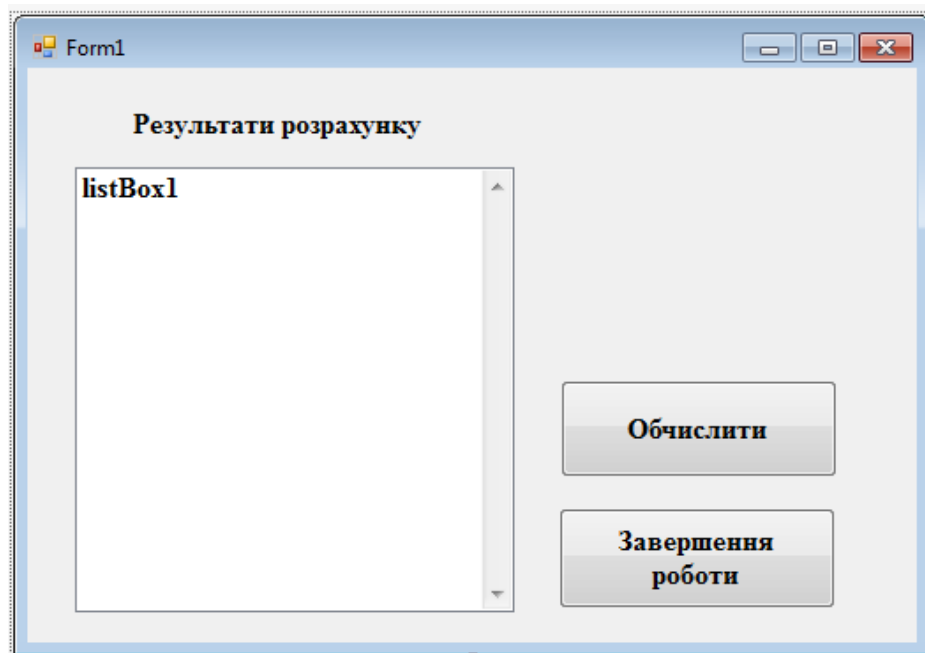


Рисунок 3.2 – Форма до задачі 3.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WAZ
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```



```

private void cmdStart_Click(object sender, EventArgs e)
{
    double x, y;          // аргумент і значення функції
    double xn = -2;       // початкове значення змінної x
    double xk = 2;        // кінцеве значення змінної x
    double dx = 0.5;      // крок зміни змінної
    listBox1.Items.Add("-----");
    listBox1.Items.Add("      x              y      ");
    listBox1.Items.Add("-----");
    for (x = xn; x <= xk; x = x + dx)
    {
        y = -2.4 * x * x + 5 * x - 3;
        listBox1.Items.Add($"{x,8:F2}          {y,8:F3} ");
    }
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунків подано на рис. 3.3.

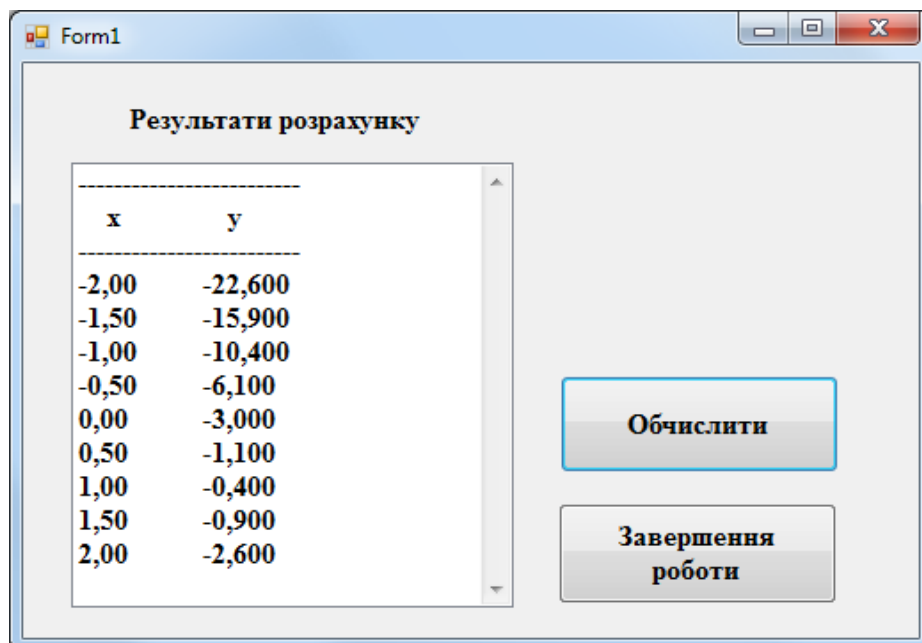


Рисунок 3.3 – Результати розв’язання задачі 3.1

Пояснення:

1)listBox1.Items.Add(str);
Виведення даних у поле listBox.

Тема 4. ОДНОВИМІРНІ МАСИВИ

Мета роботи: вивчити елементи керування — мітку `Label`, кнопку `Button`, поле введення `TextBox`, список `ListBox`, комбіноване вікно `ComboBox`.

Задача 4.1

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

Створимо Windows-додаток для розв'язання цієї задачі.

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

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 4.1, 4.2.

На формі розташуйте:

- одне поле списку (`ListBox`),
- дві командні кнопки (`Button`).

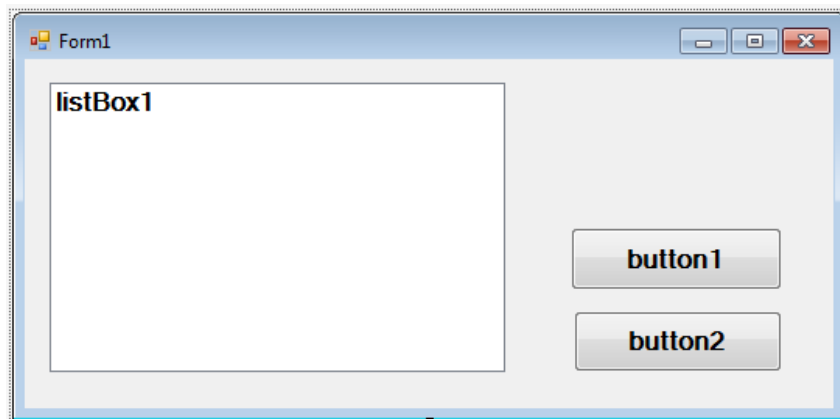


Рисунок 4.1 – Форма до задачі 4.1

2. Установіть значення властивостей об'єктів згідно з табл. 4.1.

Таблиця 4.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Поле списку	ListBox1		
Командна кнопка	Button1	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

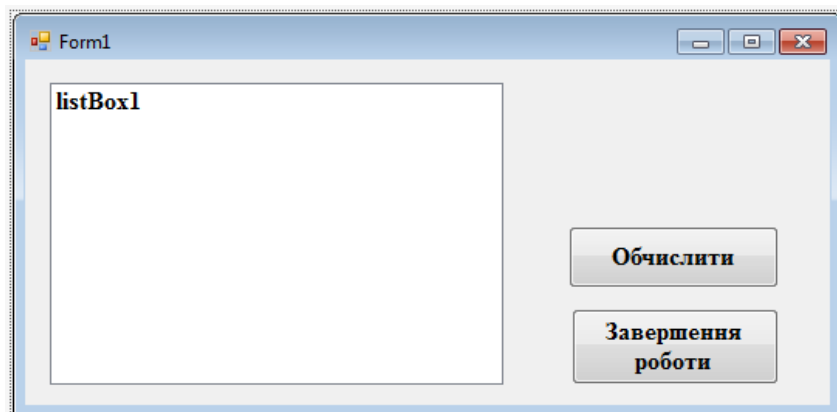


Рисунок 4.2 – Форма до задачі 4.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA11
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void cmdStart_Click(object sender, EventArgs e)
            {
                const int n = 12; // кількість елементів масиву
                double[] A = new double[n];
                int i;
                double s;
                // Обчислення і виведення елементів масиву
                listBox1.Items.Add(" Масив A: ");
                for (i = 0; i < n; i++)
                {
                    A[i] = 12 + i / Math.Sin(i + 1);
                    listBox1.Items.Add($" A[{i}]= {A[i]:F4}");
                }
                listBox1.Items.Add(" ");
                // Обчислення суми елементів масиву з 5-го до 7-го
                listBox1.Items.Add("Сума елементів масиву з 5-го до 7-го");
                s = 0;
                for (i = 5; i <= 7; i++)
                    s = s + A[i];
                listBox1.Items.Add($"s= {s:F4} ");
                listBox1.Items.Add(" ");
                //Виведення елементів масиву у зворотному порядку з 7-го до 5-го
                listBox1.Items.Add("Виведення елементів масиву у
                                зворотному порядку з 7 - го до 5 - го");
                for (i = 7; i >= 5; i--)
                    listBox1.Items.Add($"A[{i}]= {A[i]:F4}");
            }
        }
    }
}
```

```

        listBox1.Items.Add(" ");
        //Виведення парних за індексом елементів масиву
        listBox1.Items.Add("Виведення парних за індексом елементів масиву");
        for (i = 2; i <= 10; i = i + 2)
            listBox1.Items.Add($"A[{i}]= {A[i]:F4}");
        listBox1.Items.Add(" ");
    }

    private void cmdEnd_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}

```

Результати розрахунку подано на рис. 4.3.

Form1

Масив А:
A[0]= 12,0000
A[1]= 13,0998
A[2]= 26,1723
A[3]= 8,0360
A[4]= 7,8287
A[5]= -5,8945
A[6]= 21,1326
A[7]= 19,0753
A[8]= 31,4119
A[9]= -4,5435
A[10]= 1,9999
A[11]= -8,5005

Сума елементів масиву з 5-го до 7-го
s= 34,3134

Виведення елементів масиву у зворотному порядку з 7-го до 5-го
A[7]= 19,0753
A[6]= 21,1326
A[5]= -5,8945

Виведення парних за індексом елементів масиву
A[2]= 26,1723
A[4]= 7,8287
A[6]= 21,1326
A[8]= 31,4119
A[10]= 1,9999

Обчислити

Завершення роботи

Рисунок 4.3 – Результати розв’язання задачі 4.1

Задача 4.2. У масиві цілих чисел $A(15)$ замінити значення від’ємних елементів квадратами значень, значення додатних елементів збільшити на 7, а нульові елементи залишити без зміни.

Вивести новий масив A .

ПОРЯДОК ДІЙ

1. Розташуйте на формі об’єкти, як показано на рис. 4.4, 4.5.

На формі розташувати:

- дві мітки (Label),
- два списки (ListBox),
- три командні кнопки (Button).

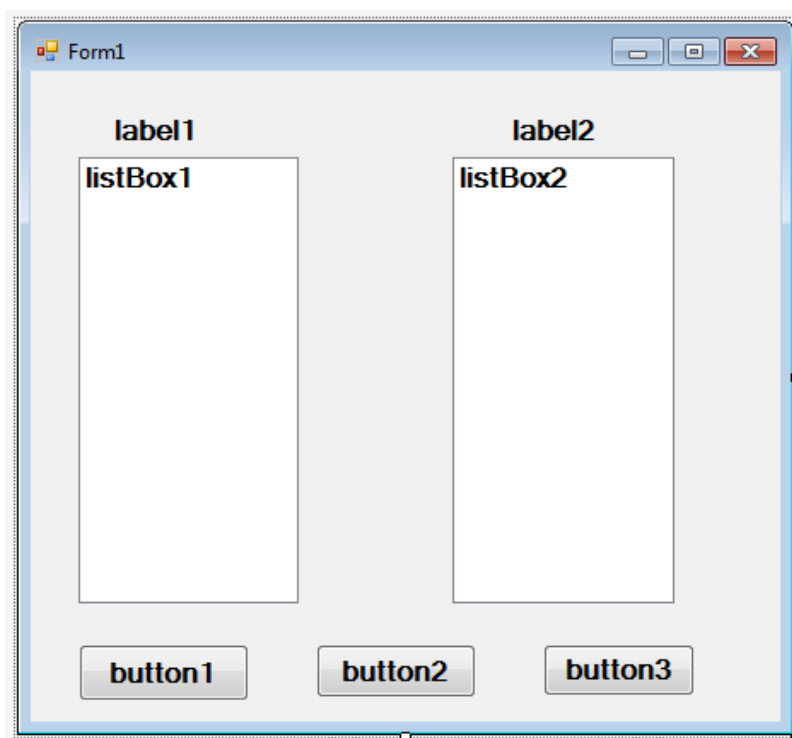


Рисунок 4.4 – Форма до задачі 4.2

2. Установіть значення властивостей об’єктів згідно з табл. 4.2.

Таблиця 4.2 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1	Text	Одновимірні масиви
Мітка	Label1	Text	Початковий масив:
Мітка	Label2	Text	Новий масив:
Список	ListBox1		
Список	ListBox2		
Командна кнопка	Button1	Name	Button1
		Text	Заповнити масив
Командна кнопка	Button2	Name	Button2
		Text	Заміна елементів
Командна кнопка	Button3	Name	Button3
		Text	Завершення роботи

The image shows a Windows application window with the title bar 'Одновимірні масиви'. Inside the window, there are two labels: 'Початковий масив:' and 'Новий масив:'. Below each label is a list box, 'listBox1' and 'listBox2' respectively. At the bottom of the window, there are three buttons: 'Заповнити масив', 'Заміна елементів', and 'Завершення роботи'.

Рисунок 4.5 – Форма до задачі 4.2

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Заповнити масив>, *клацання мишею по командній кнопці* <Заміна елементів> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA26
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // Глобальна змінна, видима всім методам
            int[] A = new int[15];

            // Заповнення початкового масиву
            private void button1_Click(object sender, EventArgs e)
            {
                // Очищуємо елемент керування
                listBox1.Items.Clear();
                // Ініціалізуємо клас випадкових чисел
                Random rnd = new Random();
                // Генеруємо та виводимо 15 елементів
                for (int i = 0; i < 15; i++)
                {
                    A[i] = rnd.Next(-50, 50);
                    listBox1.Items.Add($"A[{i}] = {A[i]}");
                }
            }

            // Заміна елементів
            private void button2_Click(object sender, EventArgs e)
            {
                // Очищуємо елемент керування
                listBox2.Items.Clear();
                // Оброблюємо всі елементи
                for (int i = 0; i < 15; i++)
                {
```



```

        if (A[i] < 0)
            A[i] = A[i] * A[i];
        else if (A[i] > 0)
            A[i] = A[i] + 7;
        else
            A[i] = A[i];
        listBox2.Items.Add($"A[{i}]= {A[i]}");
    }
}

private void button3_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 4.6.

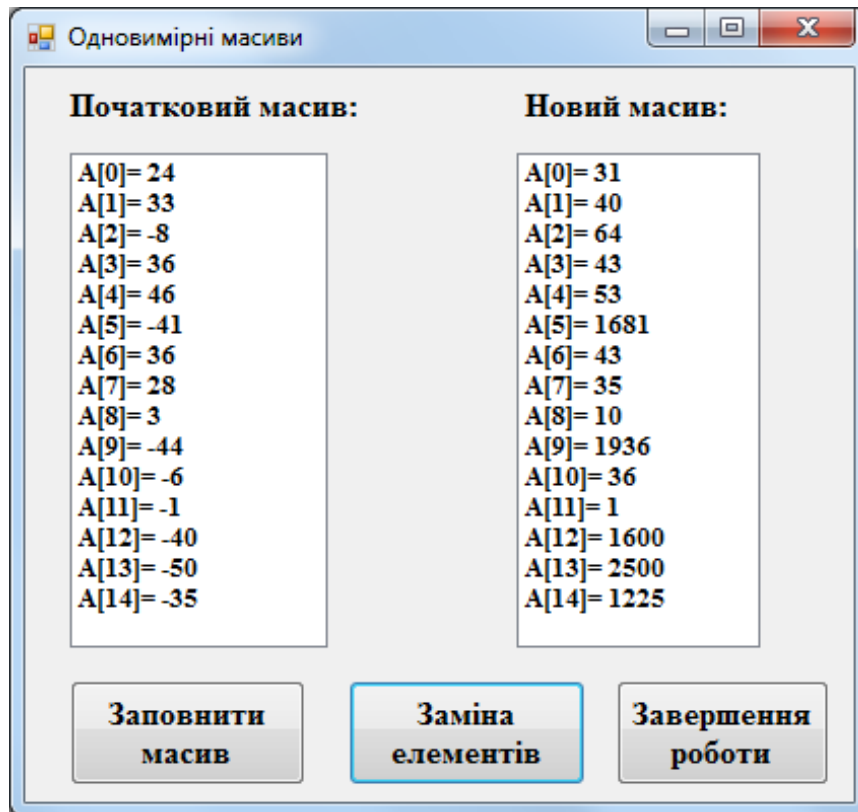


Рисунок 4.6 – Результати розв’язання задачі 4.2

Пояснення:

1) `int[] A = new int[15];`

Оголошується масив A з 15 цілих чисел типу `int` і одночасно всі елементи масиву автоматично (неявно) ініціалізуються нулями.

2) `listBox1.Items.Add($"A[{i}]= {A[i]}");`

Виведення елементів масиву A в поле `listBox1`.

Задача 4.3. Створити *Windows-додаток*, що дозволяє:

- 1) заповнювати одновимірний масив $A(n)$ двома способами:
 - випадковими дійсними числами;
 - шляхом розрахунку елементів масиву за формулою;
- 2) знаходити середнє арифметичне елементів масиву;
- 3) знаходити максимальний елемент масиву та його індекс (номер).

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти (рис. 4.7, 4.8).

На формі розташувати:

- чотири мітки (Label);
- одне текстове поле (TextBox);
- один об'єкт комбіноване вікно (ComboBox) для вибору способу заповнення масиву;
- одне поле списку (ListBox);
- чотири командні кнопки (Button).

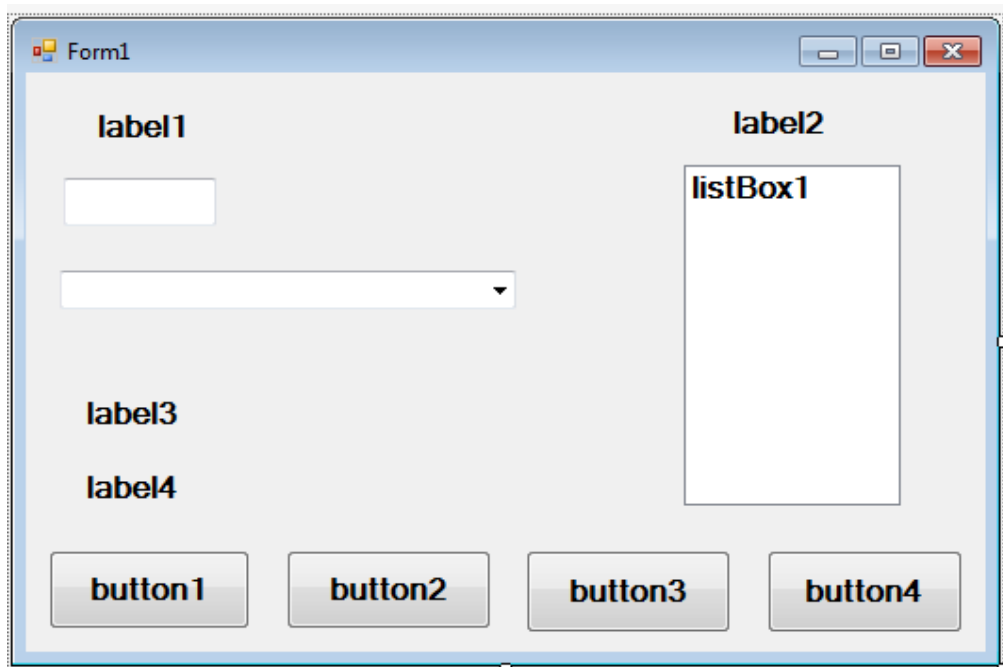


Рисунок 4.7 – Форма до задачі 4.3

2. Установіть значення властивостей об'єктів відповідно до табл. 4.3.

Таблиця 4.3 – Значення властивостей об’єктів

Об’єкт	Ім’я об’єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1		
Мітка	Label1	text	Кількість елементів масиву
Мітка	Label2	text	Елементи масиву
Мітка	Label3		
Мітка	Label4		
Текстове поле	TextBox1	Name	txtN
Поле списку	ListBox1		
Командна кнопка	Button1	Name	cmdMas
		text	Заповнення масиву
Командна кнопка	Button2	Name	cmdSred
		text	Обчислення середнього арифметичного
Командна кнопка	Button3	Name	cmdMax
		text	Обчислення максимального елемента масиву та його індексу
Командна кнопка	Button4	Name	cmdEnd
		text	Завершення роботи
Комбіноване вікно	ComboBox1	Name	ComboBox1
		Text	Виберіть спосіб заповнення масиву
		Items	Випадковими числами Розрахувати за формулою

Рисунок 4.8 – Форма до задачі 4.3

3. Програмування

1) Напишіть оброблювач подій *клацання мишею по командній кнопці*
<Заповнення масиву>.

2) Напишіть оброблювач подій *клацання мишею по командній кнопці*
<Обчислення середнього арифметичного>.

3) Напишіть оброблювач подій *клацання мишею по командній кнопці*
<Обчислення максимального елемента масиву та його індексу>.

4) Напишіть оброблювач подій *клацання мишею по командній кнопці*
<Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA7
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            int n;
            double[] A;

            private void cmdMas_Click(object sender, EventArgs e)
            {
                int i;
                Random rnd = new Random();
                n = Convert.ToInt32(txtN.Text);
                A = new double[n];
                switch (comboBox1.SelectedIndex)
                {
                    case 0:
                        for (i = 0; i < n; i++)
                        {
                            A[i] = rnd.NextDouble() * 10;
                            listBox1.Items.Add($" {A[i]:F2}");
                        }
                        break;
                    case 1:
                        for (i = 0; i < n; i++)
                        {
                            A[i] = (i + 2) / 3.0;
                        }
                }
            }
        }
    }
}
```

```

        listBox1.Items.Add($" {A[i]:F2}");
    }
    break;
}
}

private void cmdSred_Click(object sender, EventArgs e)
{
    double s; // сума елементів масиву
    double sr; // середнє арифметичне елементів масиву
    int i;
    s = 0;
    for (i = 0; i < n; i++)
    {
        s = s + A[i];
    }
    sr = s / n;
    label3.Text = $"Середнє арифметичне= {sr:F3}";
}

private void cmdMax_Click(object sender, EventArgs e)
{
    double max; // максимальний елемент масиву
    int imax; // індекс максимального елемента
    int i;
    max = A[0]; imax = 0;
    for (i = 1; i < n; i++)
    {
        if (A[i] > max)
        {
            max = A[i];
            imax = i;
        }
    }
    label4.Text = $"Максимальний елемент = {max:F3} \n
        Індекс макс. елемента = {imax}";
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 4.9, 4.10.

1 спосіб заповнення масиву – випадковими числами

Form1

Кількість елементів масиву

10

Випадковими числами

Середнє арифметичне= 6,137

Максимальний елемент = 9,630

Індекс макс. елемента = 8

Елементи масиву

- 7,74
- 4,06
- 6,50
- 7,36
- 5,42
- 9,25
- 3,55
- 5,36
- 9,63
- 2,50

Заповнення масиву

Обчислення середнього арифметичного

Обчислення максимального елемента масиву та його індексу

Завершення роботи

Рисунок 4.9 – Результати розв’язання задачі 4.3

2 спосіб заповнення масиву – розрахувати за формулою

Form1

Кількість елементів масиву

10

Розрахувати за формулою

Середнє арифметичне= 2,167

Максимальний елемент = 3,667

Індекс макс. елемента = 9

Елементи масиву

- 0,67
- 1,00
- 1,33
- 1,67
- 2,00
- 2,33
- 2,67
- 3,00
- 3,33
- 3,67

Заповнення масиву

Обчислення середнього арифметичного

Обчислення максимального елемента масиву та його індексу

Завершення роботи

Рисунок 4.10 – Результати розв’язання задачі 4.3

Задача 4.4. Обчислити значення виразу:

$$P = \sum_{i=0}^9 (x_i + 5,6y_i),$$

де x_i – елемент одновимірного масиву $X(10)$; $i=0,1,\dots,9$;

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

Створити *Windows-додаток*.

Блок-схему розв’язання задачі наведено на рис. 4.14.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об’єкти, як показано на рис. 4.11, 4.12.

На формі розташувати:

- три мітки (Label),
- два поля списку (ListBox),
- три командні кнопки (Button).

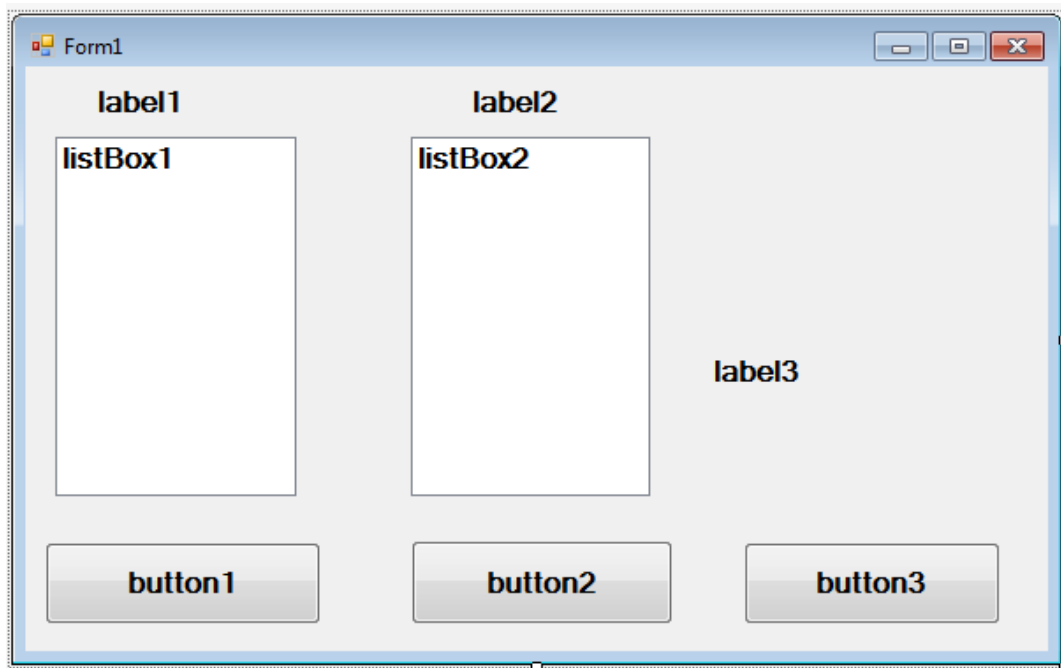


Рисунок 4.11 – Форма до задачі 4.4

2. Установіть значення властивостей об’єктів відповідно до табл. 4.4.

Таблиця 4.4 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1		
Мітка	Label1	text	Масив X
Мітка	Label2	text	Масив Y
Мітка	Label3		
Поле списку	ListBox1		
Поле списку	ListBox2		
Командна кнопка	Button1	Name	cmdMas
		text	Обчислення і виведення елементів масивів X і Y
Командна кнопка	Button2	Name	cmdP
		text	Обчислення P
Командна кнопка	Button3	Name	cmdEnd
		text	Завершення роботи

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two list boxes on the left, labeled "listBox1" and "listBox2", each preceded by the heading "Масив X" and "Масив Y" respectively. To the right of these list boxes is a label "label3". At the bottom of the window, there are three buttons arranged horizontally. The first button on the left is labeled "Обчислення і виведення елементів масивів X і Y". The middle button is labeled "Обчислення P". The button on the right is labeled "Завершення роботи".

Рисунок 4.12 – Форма до задачі 4.4

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислення і виведення елементів масивів X і Y>, *клацання мишею по командній кнопці* <Обчислення P>, *клацання мишею по командній кнопці* <Завершення роботи>

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA9
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            const int n = 10; // кількість елементів кожного масиву
            double[] X = new double[n];
            double[] Y = new double[n];

            private void cmdMas_Click(object sender, EventArgs e)
            {
                //Обчислення і виведення елементів масиву X
                for (int i = 0; i < n; i++)
                {
                    X[i] = i / 3.0;
                    listBox1.Items.Add($"{X[i]:F3}");
                }
                // Обчислення і виведення елементів масиву Y
                for (int i = 0; i < n; i++)
                {
                    Y[i] = i / 2.0 + 4.0;
                    listBox2.Items.Add($"{Y[i]:F3}");
                }
            }
        }
    }
}
```

```

private void cmdP_Click(object sender, EventArgs e)
{
    double s, p;
    //Накопичення суми
    s = 0;
    for (int i = 0; i < n; i++)
    {
        s = s + (X[i] + 5.6 * Y[i]);
    }
    p = s;
    label3.Text = $"p= {p:F3}";
}

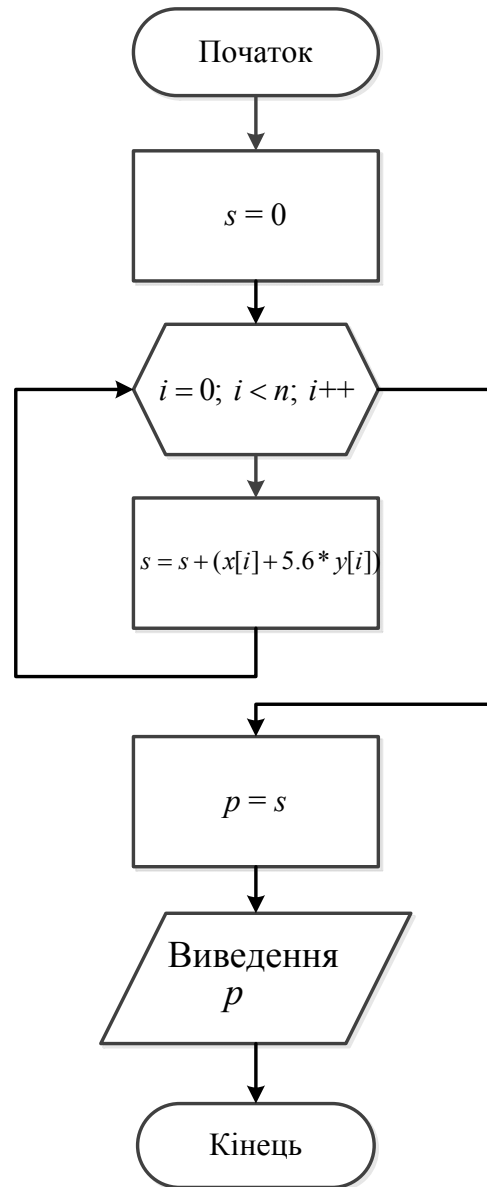
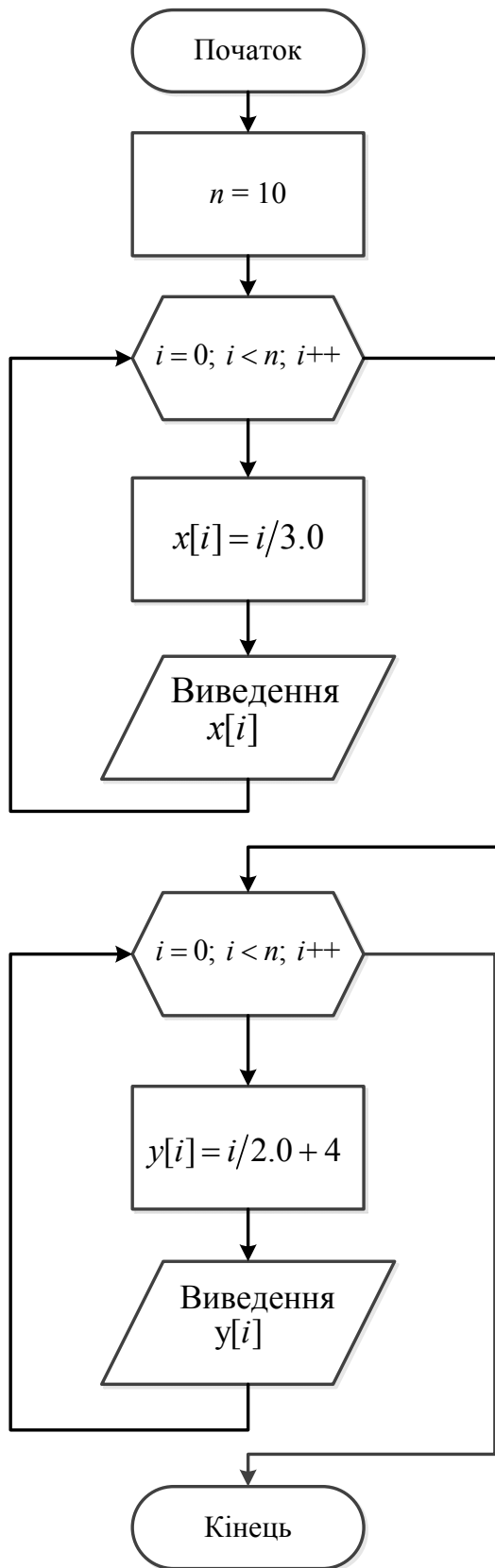
private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 4.13.

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two vertical lists of numbers. The first list, labeled "Масив X", contains the following values: 0,000, 0,333, 0,667, 1,000, 1,333, 1,667, 2,000, 2,333, 2,667, and 3,000. The second list, labeled "Масив Y", contains the following values: 4,000, 4,500, 5,000, 5,500, 6,000, 6,500, 7,000, 7,500, 8,000, and 8,500. To the right of these lists, the text "p= 365,000" is displayed. At the bottom of the window, there are three buttons: "Обчислення і виведення елементів масивів X і Y", "Обчислення P", and "Завершення роботи".

Рисунок 4.13 – Результати розв’язання задачі 4.4



Блок-схеми оброблювачів подій cmdMas_Click і cmdP_Click

Рисунок 4.14 – Блок-схема алгоритму до задачі 4.4

Задача 4.5. Обчислити значення елементів одновимірного масиву $Q(10)$ за формулою:

$$q_i = e^x (a_i - b_i)^2 + k ,$$

де $x = 2$; $k = 3,5$;

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

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

q_i – елемент одному. масиву $Q(10)$ $i = 0,1,\dots,9$.

Значення елементів масивів A і B задати в програмному коді. Створити *Windows-додаток*. Блок-схема обробника подій `cmdStart_Click` подана на рис. 4.18.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 4.15, 4.16.

На формі розташувати:

- три мітки (Label),
- три поля списку (ListBox),
- дві командних кнопки (Button).

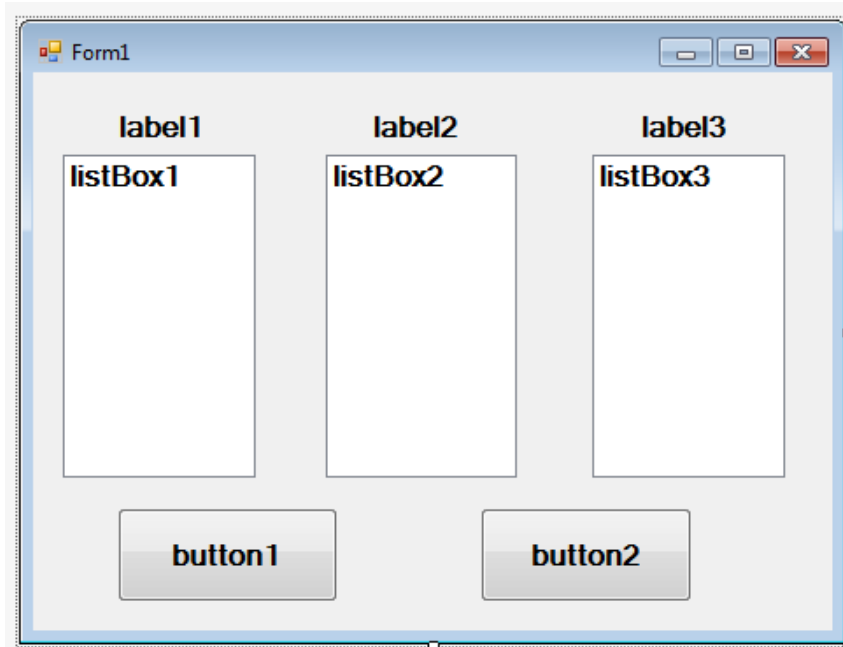


Рисунок 4.15 – Форма до задачі 4.5

2. Установіть значення властивостей об'єктів відповідно до табл. 4.5.

Таблиця 4.5 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Мітка	Label1	text	Масив А
Мітка	Label2	text	Масив В
Мітка	Label3	text	Масив Q
Поле списку	ListBox1		
Поле списку	ListBox2		
Поле списку	ListBox3		
Командна кнопка	Button1	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

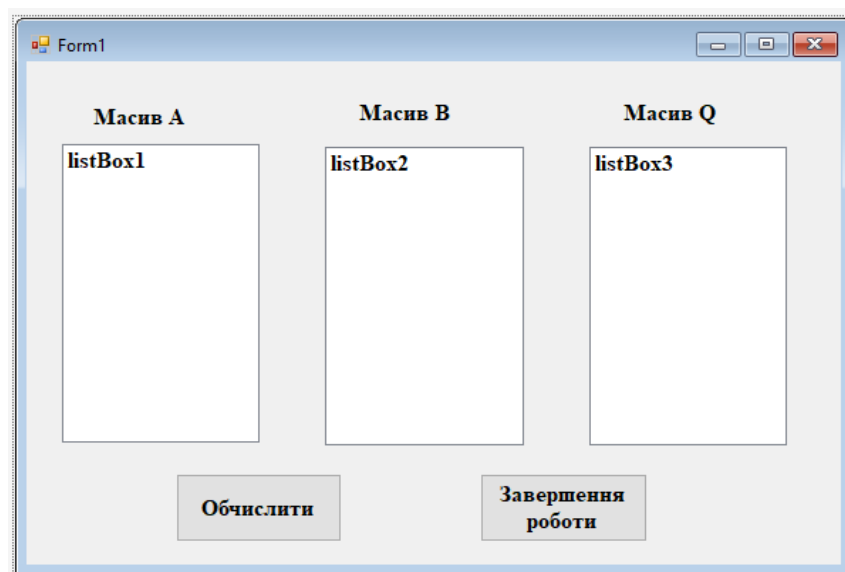


Рисунок 4.16 – Форма до задачі 4.5

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> та *клацання мишею по командній кнопці* <Завершення роботи>

Програмний код:

```
using System;
using System.Windows.Forms;

namespace W8A
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

private void cmdStart_Click(object sender, EventArgs e)
{
    const int n = 10; //кількість елементів кожного масиву
    const double x = 2.0, k = 3.5;
    double[] A = new double[n] { 1.0, 3.0, 5.0, 7.0, 8.0,
                                   9.0, 10.0, 11.0, 13.0, 15.0 };
    double[] B = new double[n] { 2.0, 5.0, 7.0, 8.0, 9.0,
                                   12.0, 14.0, 15.0, 17.0, 18.0 };
    double[] Q = new double[n];
    //Обчислення елементів масиву Q і виведення масивів
    //в поля списків
    for (int i = 0; i < n; i++)
    {
        Q[i] = Math.Exp(x) * Math.Pow((A[i] - B[i]), 2) + k;
        listBox1.Items.Add($"{A[i]:F3}");
        listBox2.Items.Add($"{B[i]:F3}");
        listBox3.Items.Add($"{Q[i]:F3}");
    }
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 4.17.

Масив А	Масив В	Масив Q
1,000	2,000	10,889
3,000	5,000	33,056
5,000	7,000	33,056
7,000	8,000	10,889
8,000	9,000	10,889
9,000	12,000	70,002
10,000	14,000	121,725
11,000	15,000	121,725
13,000	17,000	121,725
15,000	18,000	70,002

Обчислити Завершення роботи

Рисунок 4.17 – Результати розв’язання задачі 4.5

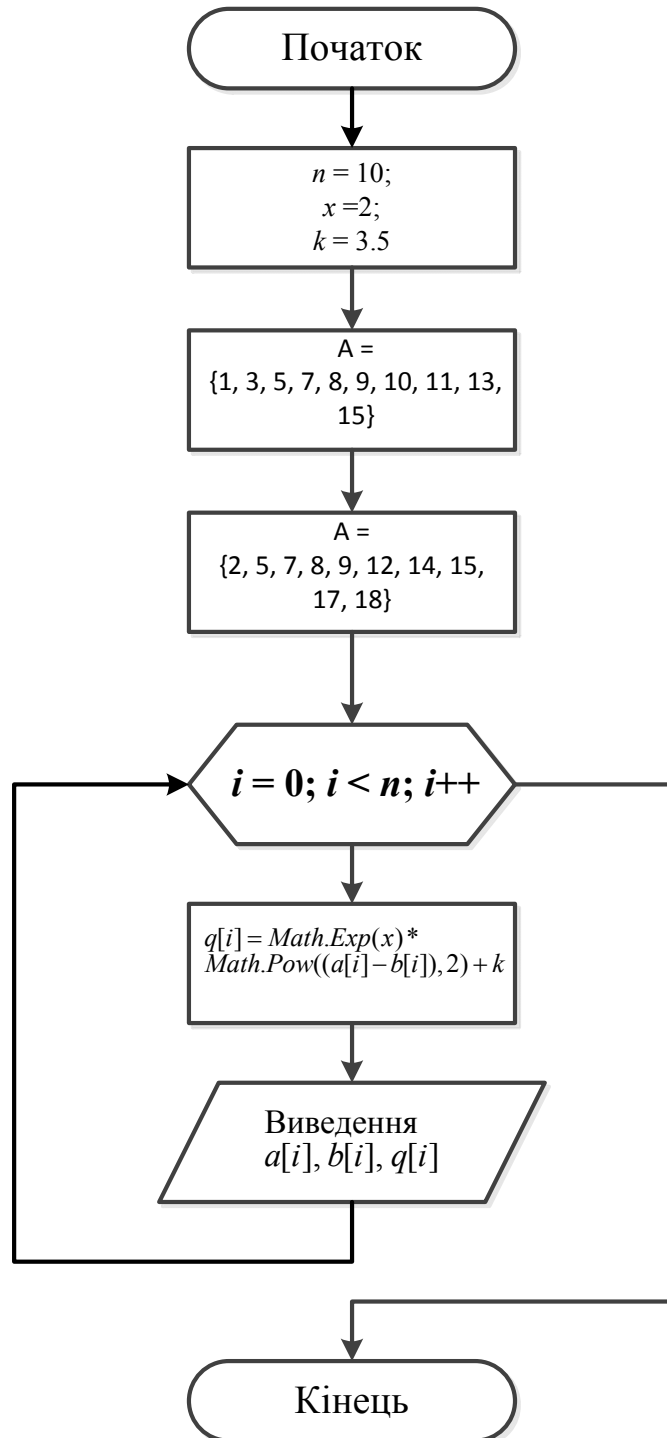


Рисунок 4.18 – Блок-схема алгоритму до задачі 4.5

Задача 4.6. Обчислити значення виразу:

$$F = c \cdot 3 \sqrt[3]{\prod_{i=0}^9 (x_i + a)},$$

де $a = 7,5$; $c = 2.45$; x_i – елемент одновимірного масиву $X(10)$; $i = 0, 1, \dots, 9$.

Створити *Windows-додаток*. Блок-схему оброблювача подій `cmdStart_Click` наведено на рис. 4.19.

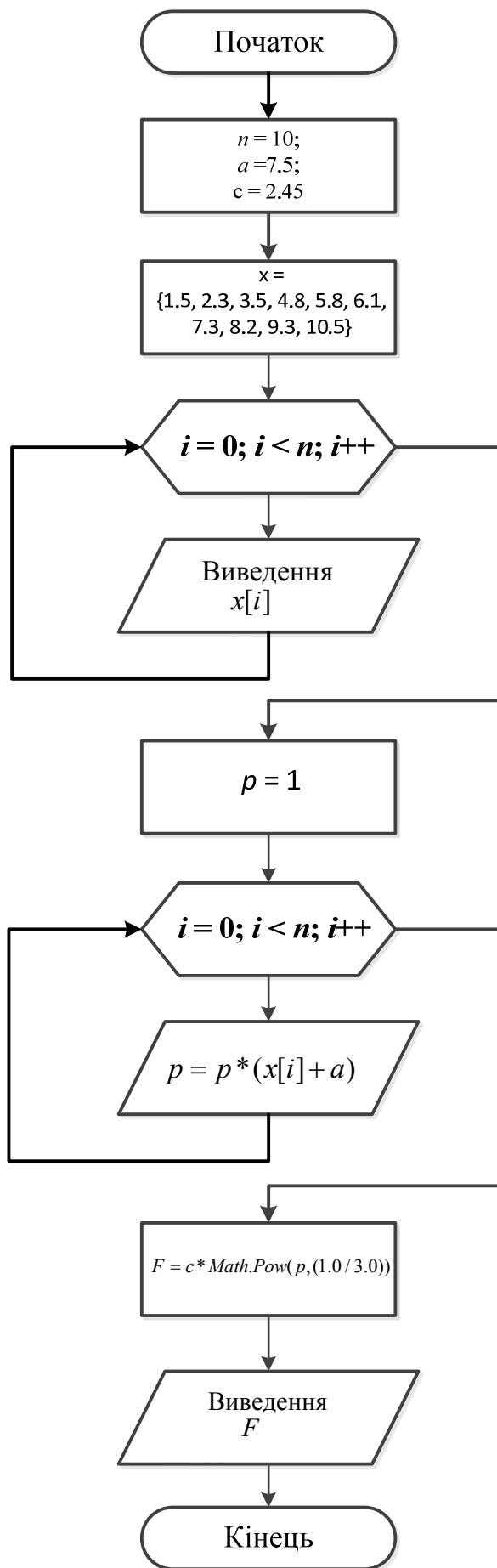


Рисунок 4.19 – Блок-схема алгоритму до задачі 4.6

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 4.20, 4.21.

На формі розташувати:

- дві мітки (Label),
- одне поле списку (ListBox)
- дві командних кнопки (Button).

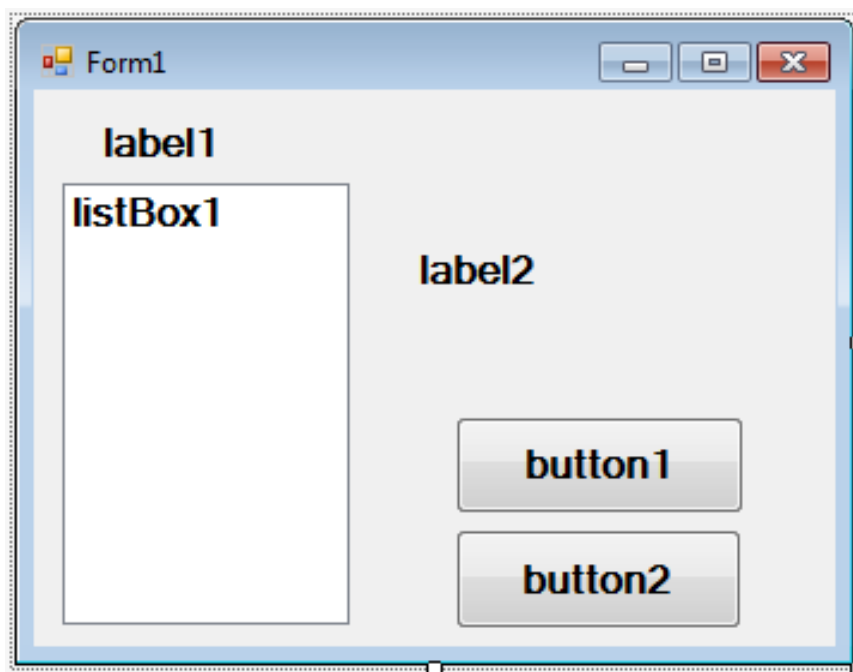


Рисунок 4.20 – Форма до задачі 4.6

2. Установіть значення властивостей об'єктів відповідно до табл. 4.6.

Таблиця 4.6 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Метка	Label1	text	Масив X
Метка	Label2		
Поле списку	ListBox1		
Командная кнопка	Button1	Name	cmdStart
		text	Обчислити
Командная кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

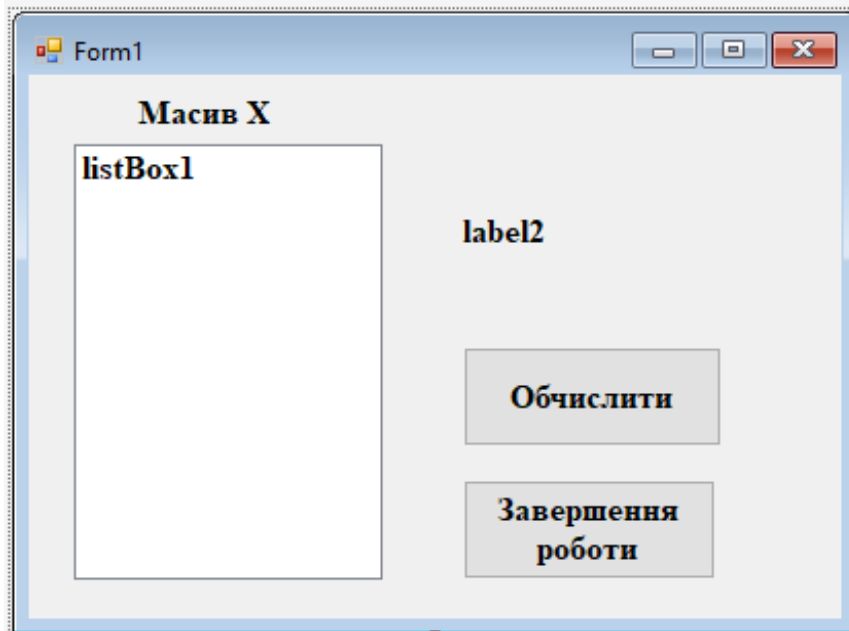


Рисунок 4.21 – Форма до задачі 4.6

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> та *клацання мишею по командній кнопці* <Завершення роботи>

Програмний код:

```
using System;
using System.Windows.Forms;

namespace W10A
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            const int n = 10; // кількість елементів масиву
            const double a = 7.5, c = 2.45;
            int i;
            double p, F;
```

```

double[] X = new double[n] { 1.5, 2.3, 3.5, 4.8, 5.8,
                              6.1, 7.3, 8.2, 9.3, 10.5 };
// Виведення елементів масиву
for (i = 0; i < n; i++)
    listBox1.Items.Add(X[i]);
// Накопичення суми елементів з 4-го по 10-ий
p = 1;
for (i = 0; i < n; i++)
    p = p * (X[i] + a);
F = c * Math.Pow(p, (1.0 / 3.0));
label2.Text = $"F= {F:F3}";
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 4.22.

Рисунок 4.22 – Результати розв'язання задачі 4.6

Тема 5. ДВОВИМІРНІ МАСИВИ

Мета роботи: вивчити елементи керування – мітку **Label**, кнопку **Button**, поле введення **TextBox**, список **ListBox**, компонент для подання таблиці на формі **DataGridView**.

Задача 5.1

Створити двовимірний масив A розміром 7×7 ,

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

Елементи масиву заповнити випадковими числами.

Розробити оброблювач кнопки, який буде шукати максимальний елемент на головній діагоналі матриці. Результат розрахунку вивести на мітку. Для розв'язання задачі створити Windows-додаток.

Спосіб 1. Елементи матриці вивести на компонент **DataGridView**. Цей компонент служить для подання таблиці (по суті двовимірного масиву) на формі. Блок-схему алгоритму подано на рис. 5.4.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти (рис. 5.1, 5.2).

На формі розташувати:

- дві мітки (**Label**);
- одне текстове поле (**TextBox**);
- один компонент для подання таблиці на формі (**DataGridView**);
- дві командні кнопки (**Button**).

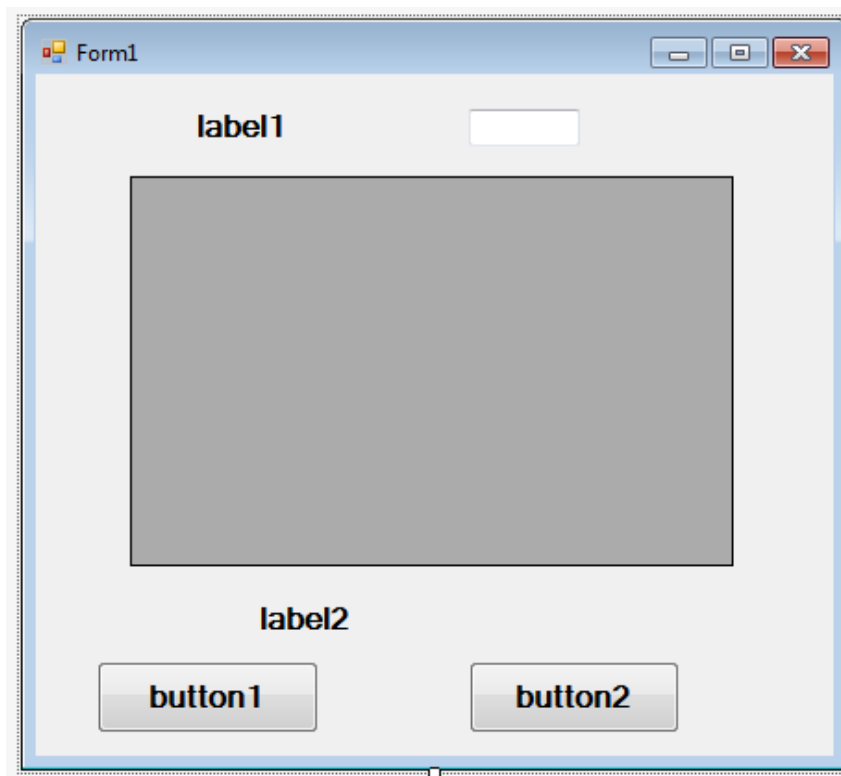


Рисунок 5.1 – Форма до задачі 5.1

2. Установіть значення властивостей об'єктів відповідно до табл. 5.1.

Таблиця 5.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1		
Мітка	Label1	text	Кількість рядків і стовпців n =
Мітка	Label2		
Текстове поле	TextBox1	Name	txtn
Компонент для подання таблиці на формі	DataGridView1		
Командна кнопка	Button1	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

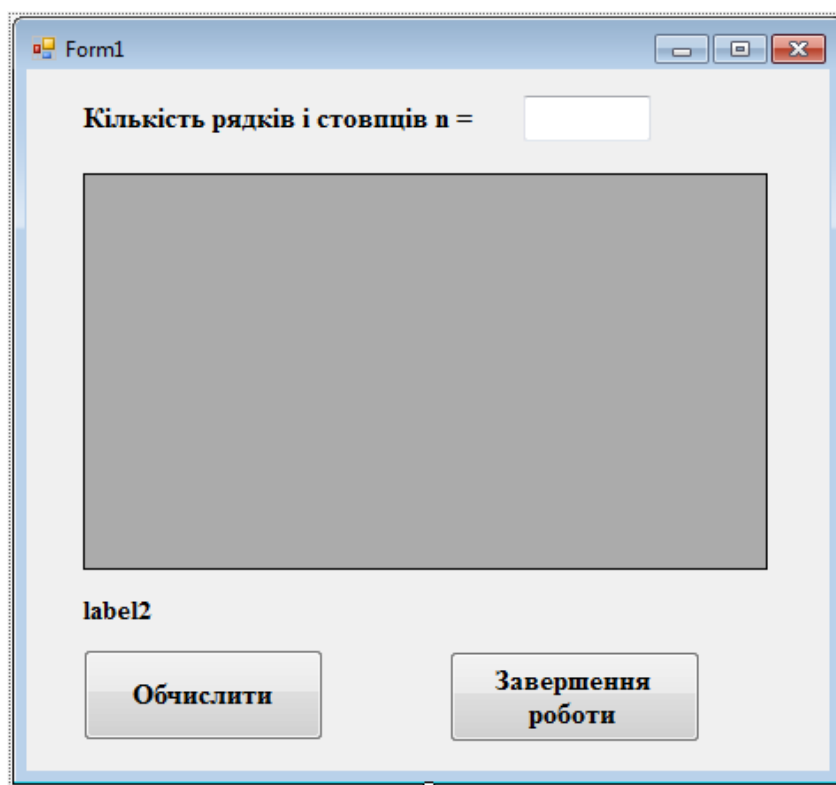


Рисунок 5.2 – Форма до задачі 5.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці <Обчислити>* і *клацання мишею по командній кнопці <Завершення роботи>*.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA27A
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void cmdStart_Click(object sender, EventArgs e)
            {
                int n; // кількість рядків і стовпців матриці
                n = Convert.ToInt32(txtn.Text);
                // Оголошення масиву
                int[,] A = new int[n, n];
                //Заповнюємо матрицю випадковими числами
                Random rnd = new Random();
                for (int i = 0; i < n; i++)
                {
                    for (int j = 0; j < n; j++)
                    {
                        A[i, j] = rnd.Next(-50, 50);
                    }
                }
                // устанавлюємо кількість рядків і стовпців таблиці
                dataGridView1.RowCount = n;
                dataGridView1.ColumnCount = n;

                // устанавлюємо ширину комірки таблиці
                for (int j = 0; j < n; j++)
                    dataGridView1.Columns[j].Width = 50;

                // Виводимо матрицю в dataGridView1,
                // заповнюємо комірки таблиці елементами масиву
                for (int i = 0; i < n; i++)
                {
                    for (int j = 0; j < n; j++)
                    {
                        dataGridView1.Rows[i].Cells[j].Value = A[i, j];
                    }
                }
            }
        }
    }
}
```

```

// Пошук максимального елемента на головній діагоналі
int max = A[0, 0];
int imax = 0;
for (int i = 0; i < n; i++)
{
    if (A[i, i] > max)
    {
        max = A[i, i];
        imax = i;
    }
}

// Виводимо результат
label2.Text = $" Максимальний елемент на головній
                діагоналі A[{imax},{imax}]= {max}";
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 5.3.

Кількість рядків і стовпців n = 7

▶	24	39	-44	-35	-42	-35	6
	-43	38	35	-10	13	4	-32
	-7	-41	-17	-33	6	9	12
	-45	-7	-48	47	-32	21	25
	38	-24	-14	-31	-22	-9	34
	-2	9	-20	-2	31	33	37
*	-5	-13	12	-24	-1	21	28

Максимальний елемент на головній діагоналі A[3,3]=47

Обчислити Завершення роботи

Рисунок 5.3 – Результати розв’язання задачі 5.1 (спосіб 1)

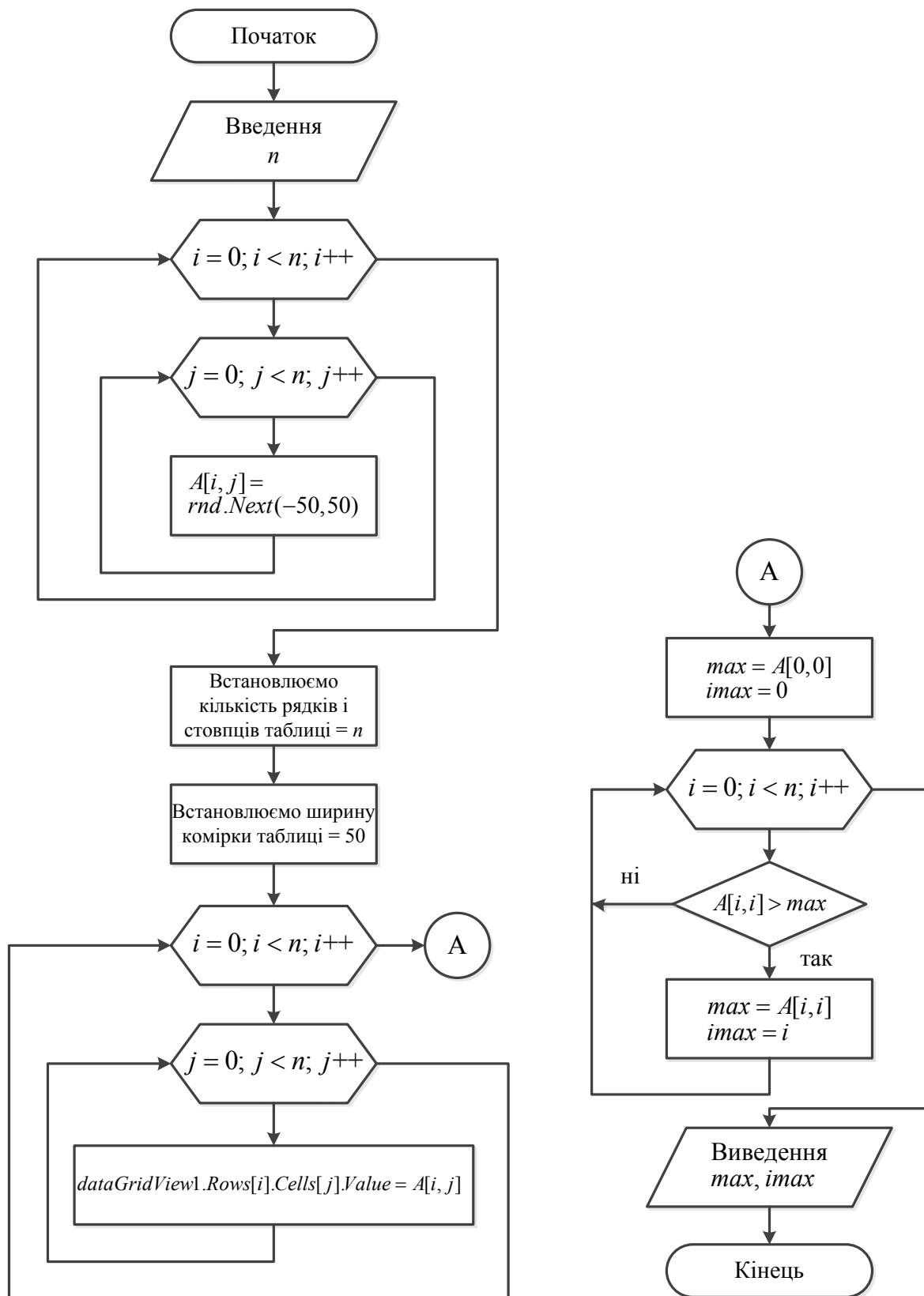


Рисунок 5.4 – Блок-схема алгоритму до задачі 5.1 (спосіб 1)

Пояснення:

При роботі із двовимірними масивами введення і виведення інформації на екран зручно організовувати у вигляді таблиць. Елемент керування DataGridView може бути використаний для відображення інформації у вигляді двовимірної таблиці. Для звертання до комірки в цьому елементі необхідно вказати номер рядка й номер стовпця, наприклад:

```
dataGridView1.Rows[2].Cells[7].Value="*";
```

Цей код запише в другий рядок і сьомий стовпець знак зірочки.

Спосіб 2. Елементи матриці задати за допомогою компонента DataGridView.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти (рис. 5.5, 5.6).

На формі розташувати:

- дві мітки (Label);
- одне текстове поле (TextBox);
- один компонент для подання таблиці на формі (DataGridView);
- три командні кнопки (Button).

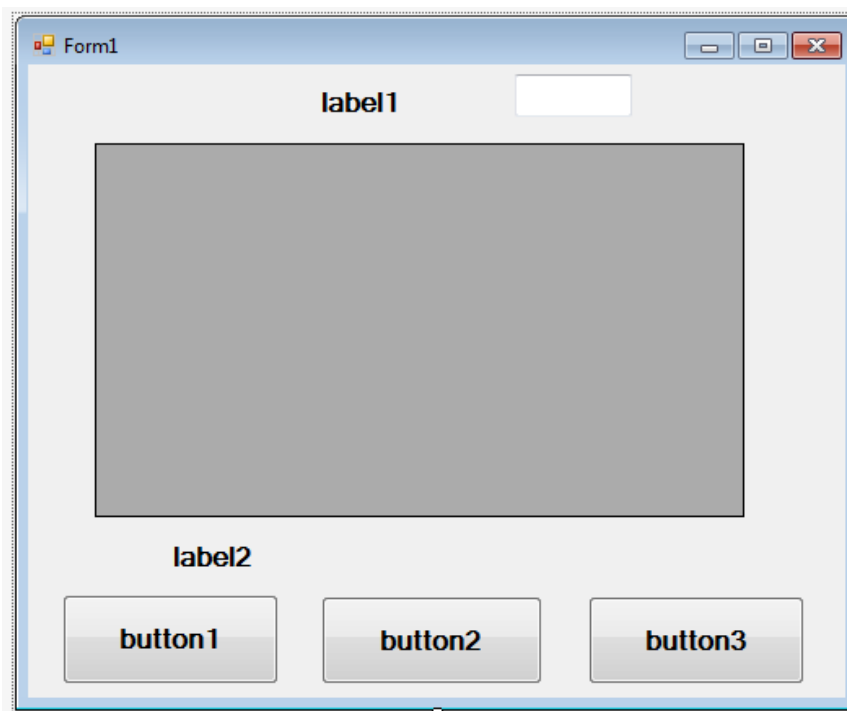


Рисунок 5.5 – Форма до задачі 5.1 (спосіб 2)

2. Установіть значення властивостей об'єктів відповідно до табл. 5.2.

Таблиця 5.2 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Мітка	Label1	text	Кількість рядків і стовпців n =
Мітка	Label2		
Текстове поле	TextBox1	Name	txtn
Компонент для подання таблиці на формі	DataGridView1		
Командна кнопка	Button1	Name	cmdMas
		text	Уведення масиву
Командна кнопка	Button2	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

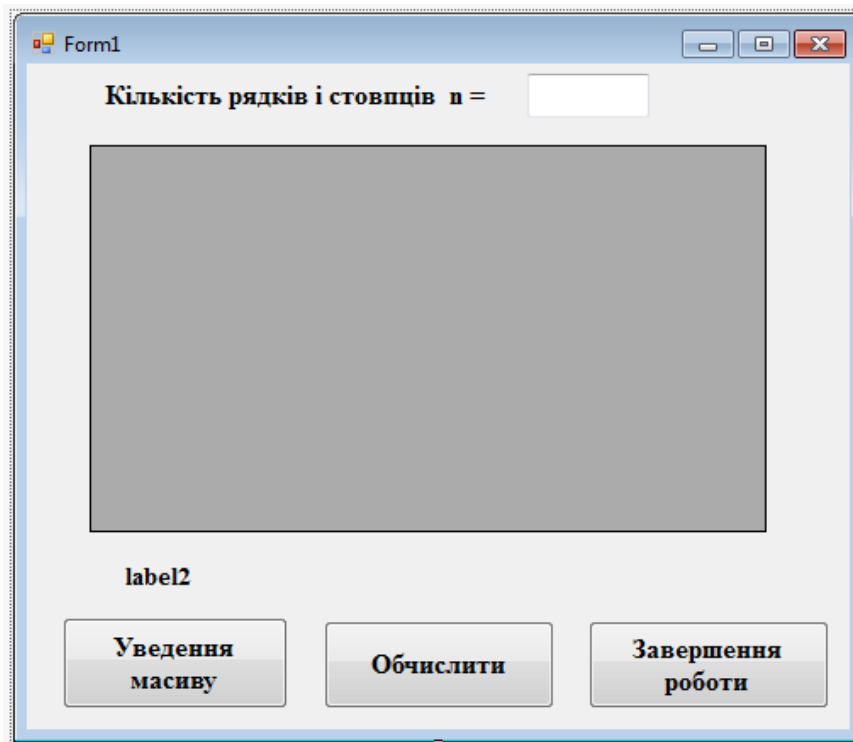


Рисунок 5.6 – Форма до задачі 5.1 (спосіб 2)

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Уведення масиву>, *клацання мишею по командній кнопці* < Обчислити >, *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA28
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // Оголошення глобальних змінних
            int n; // кількість рядків і стовпців матриці
            int[,] A;

            private void cmdMas_Click(object sender, EventArgs e)
            {
                n = Convert.ToInt32(txtn.Text);
                //установлюємо кількість рядків і стовпців таблиці
                dataGridView1.RowCount = n;
                dataGridView1.ColumnCount = n;

                // установлюємо ширину комірки таблиці
                for (int j = 0; j < n; j++)
                    dataGridView1.Columns[j].Width = 50;
            }

            private void cmdStart_Click(object sender, EventArgs e)
            {
                // Оголошення масиву
                A = new int[n, n];
                // Заповнюємо елементи масиву,
                // дані беремо із комірок таблиці
                for (int i = 0; i < n; i++)
                {
                    for (int j = 0; j < n; j++)
                    {
                        A[i, j] =
                        Convert.ToInt32(dataGridView1.Rows[i].Cells[j].Value);
                    }
                }
            }
        }
    }
}
```

```

// Пошук максимального елемента на головній діагоналі
int max = A[0, 0];
int imax = 0;
for (int i = 0; i < n; i++)
{
    if (A[i, i] > max)
    {
        max = A[i, i];
        imax = i;
    }
}

// Виводимо результат
label2.Text = $"Максимальний елемент на головній
                діагоналі A[{imax},{imax}]= {max}";
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 5.7.

Кількість рядків і стовпців n = 7

	1	-7	18	-3	5	1	2
	-2	2	-2	4	7	3	4
	3	15	5	-5	9	-4	7
	14	-7	8	6	-11	6	9
	-12	11	-1	-7	18	-7	16
	8	14	18	8	24	8	28
►	9	17	21	-10	27	16	15
*							

Максимальний елемент на головній діагоналі A[4,4]=18

Уведення масиву Обчислити Завершення роботи

Рисунок 5.7 – Результати розв’язання задачі 5.1 (спосіб 2)

Задача 5.2. Створити двовимірний масив A розміром 4×5 ,

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

Елементи масиву обчислити за формулою:

$$a_{ij} = 2i + 3j + 5$$

Обчислити добуток елементів кожного стовпця матриці.

Результат одержати у вигляді вектора D , тобто обчислити

$$d_j = \prod_{i=0}^3 a_{ij}, \text{ де } j = 0, 1, \dots, 4.$$

Для розв'язання задачі створити *Windows-додаток*.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти (рис. 5.8, 5.9).

На формі розташувати:

- чотири мітки (Label);
- два текстових поля (TextBox);
- два поля списку (ListBox);
- дві командні кнопки (Button).

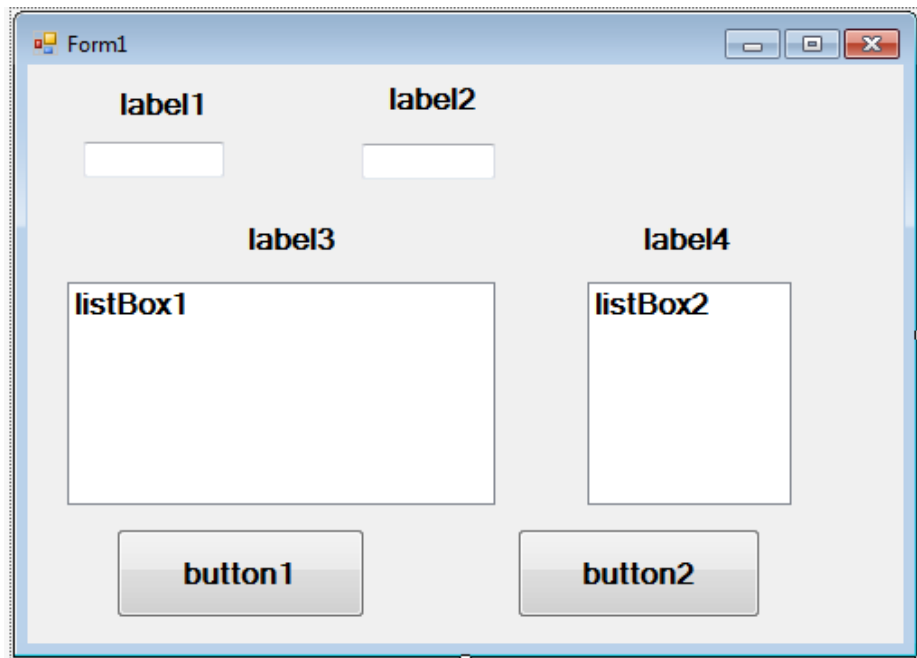


Рисунок 5.8 – Форма до задачі 5.2

2. Установіть значення властивостей об'єктів відповідно до табл. 5.3.

Таблиця 5.3 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Мітка	Label1	text	Кількість рядків
Мітка	Label2	text	Кількість стовпців
Мітка	Label3	text	Матриця A
Мітка	Label4	text	Масив D
Текстове поле	TextBox1	Name	txtN
		TextAlign	Center
Текстове поле	TextBox2	Name	txtM
		TextAlign	Center
Поле списку	ListBox1		
Поле списку	ListBox2		
Командна кнопка	Button1	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

Рисунок 5.9 – Форма до задачі 5.2

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA12AA
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            int n;      // кількість рядків матриці A
            int m;      // кількість стовпців матриці A
            int i, j;
            double p; string z;
            n = Convert.ToInt32(txtN.Text);
            m = Convert.ToInt32(txtM.Text);
            // Оголошення масивів
            double[,] A = new double[n, m];
            double[] D = new double[m];
            // Обчислення й виведення масиву A в поле списку listBox
            // z- рядок з елементів рядка масиву A
            z = " ";
            Console.WriteLine(" Масив A ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                {
                    A[i, j] = (2 * i + 3 * j + 5);
                    z = z + $"{A[i, j]:F2} ";
                }
                listBox1.Items.Add(z);
                z = " ";
            }
            // Обчислення добутку елементів стовпців масиву A
```

```

//формування й виведення масиву p у поле списку listBox2
for (j = 0; j < m; j++)
{
    p = 1;
    for (i = 0; i < n; i++)
        p = p * A[i, j];
    D[j] = p;
    listBox2.Items.Add($"{D[j]:F2}");
}
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 5.10.

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two input fields at the top: "Кількість рядків" (Number of rows) with the value "4", and "Кількість стовпців" (Number of columns) with the value "5". Below these are two text boxes. The left text box, labeled "Матриця A", contains a 4x5 grid of values: 5,00 8,00 11,00 14,00 17,00; 7,00 10,00 13,00 16,00 19,00; 9,00 12,00 15,00 18,00 21,00; 11,00 14,00 17,00 20,00 23,00. The right text box, labeled "Масив D", contains a column of five values: 3465,00; 13440,00; 36465,00; 80640,00; 156009,00. At the bottom of the window are two buttons: "Обчислити" (Calculate) and "Завершення роботи" (End work).

Рисунок 5.10 – Результати розв'язання задачі 5.2.

Тема 6. МЕТОДИ

Мета роботи: вивчити елементи керування – мітку **Label**, кнопку **Button**, поле введення **TextBox**, список **ListBox**.

Задача 6.1. Написати метод обчислення факторіала числа $K!$.
Створити *Windows-додаток*.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 6.1, 6.2.

На формі розташувати:

- три мітки (**Label**);
- одне текстове поле (**TextBox**);
- дві командні кнопки (**Button**).

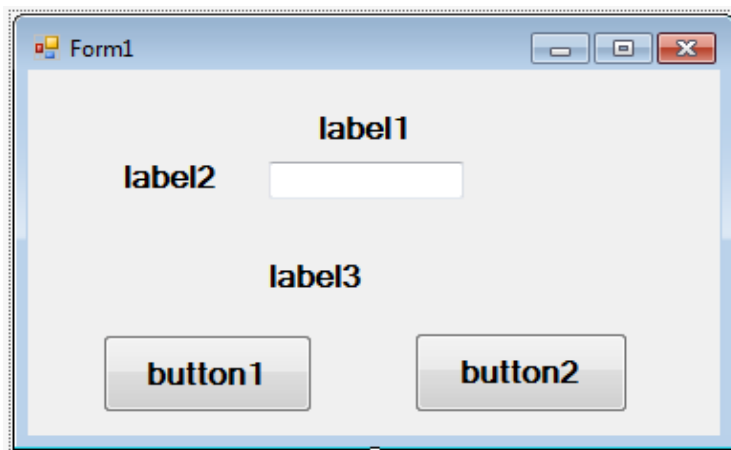


Рисунок 6.1 – Форма до задачі 6.1

2. Установіть значення властивостей об'єктів відповідно до табл. 6.1.

Таблиця 6.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Мітка	Label1	text	Обчислення факторіала числа
Мітка	Label2	text	Введіть число
Мітка	Label3		
Текстове поле	TextBox1	Name	txtN
Командна кнопка	Button1	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

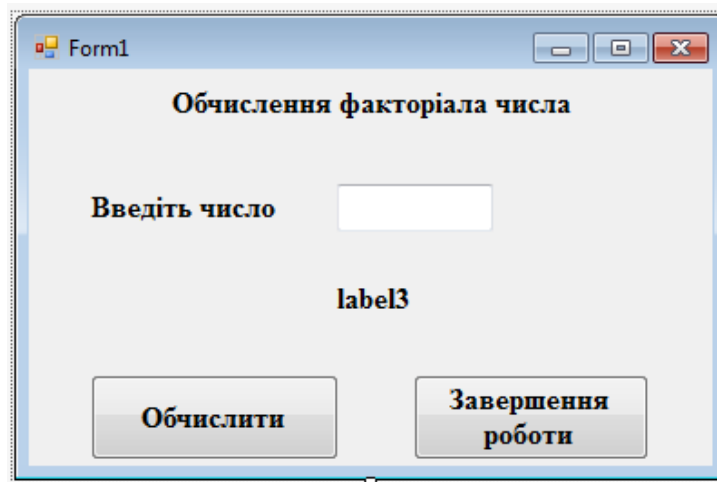


Рисунок 6.2 – Форма до задачі 6.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA13
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        static double factorial(int K)
        {
            double p = 1;
            for (int i = 2; i <= K; i++)
                p = p * i;
            return (p);
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            int N; double f;
```

```

        Console.WriteLine(" Введіть N= ");
        N = Convert.ToInt32(txtN.Text);
        f = factorial(N);
        label3.Text = $"{N}!= {f}";
    }

    private void cmdEnd_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}

```

Результати розрахунку подано на рис. 6.3.

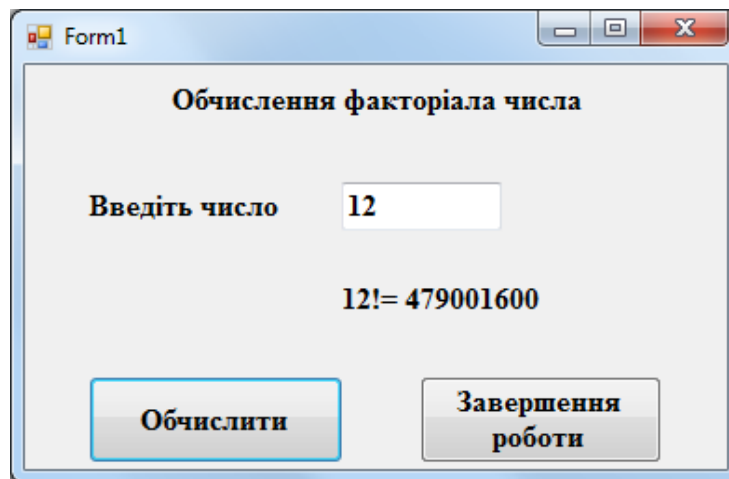


Рисунок 6.3 – Результати розв’язання задачі 6.1.

Задача 6.2. Створити метод, який підраховує кількість непарних чисел у масиві цілих чисел $A(n)$ і замінює їх на -20 .

ПОРЯДОК ДІЙ

1. Розташуйте на формі об’єкти, як показано на рис. 6.4, 6.5.

На формі розташувати:

- чотири мітки (Label),
- одне текстове поле (TextBox),
- два поля списку (ListBox),
- дві командні кнопки (Button).

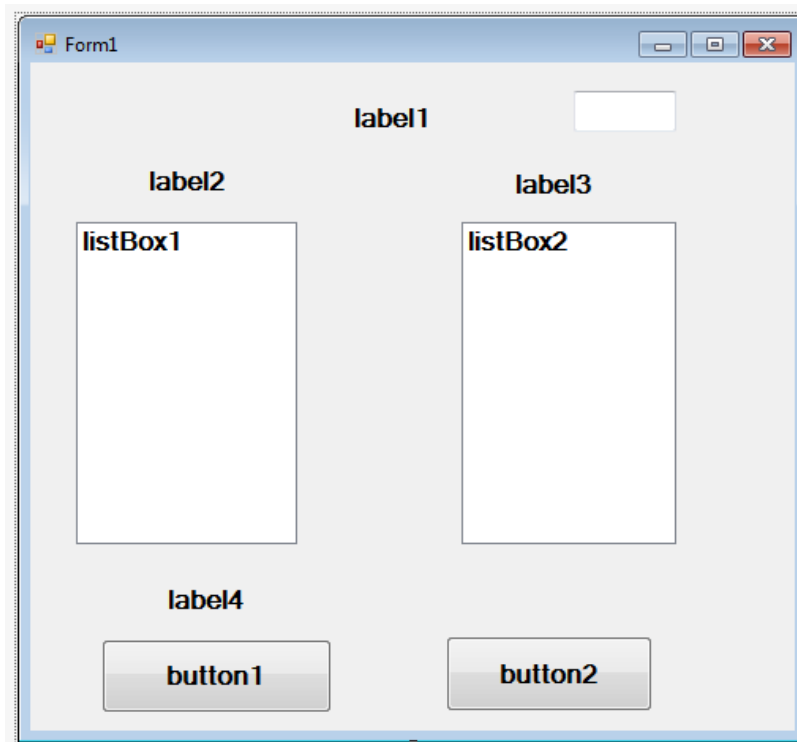


Рисунок 6.4 – Форма до задачі 6.2

2. Установіть значення властивостей об'єктів відповідно до табл. 6.2.

Таблиця 6.2 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Текстове поле	TextBox1	Name	txtn
Мітка	Label1	text	Введіть кількість елементів масиву
Мітка	Label2	text	Початковий масив
Мітка	Label3	text	Змінений масив
Мітка	Label4		
Поле списку	ListBox1		
Поле списку	ListBox2		
Командна кнопка	Button1	Name	cmdStart
		text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		text	Завершення роботи

Рисунок 6.5 – Форма до задачі 6.2

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA29
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        static void ps(int[] A, out int K)
        {
            // Метод підраховує кількість непарних чисел
            // у масиві A та замінює їх на -20
            K = 0;
        }
    }
}
```

```

        for (int i = 0; i < A.Length; i++)
        {
            if (A[i] % 2 != 0)
            {
                K = K + 1;
                A[i] = -20;
            }
        }
    }

private void cmdStart_Click(object sender, EventArgs e)
{
    int n;
    n = Convert.ToInt32(txtn.Text);
    int[] A = new int[n];
    // Створюється екземпляр rnd класу Random
    Random rnd = new Random();
    // Заповнення елементів масиву
    // та виведення їх у поле списку listBox1
    for (int i = 0; i < n; i++)
    {
        A[i] = rnd.Next(1, 10);
        listBox1.Items.Add($"{A[i]}");
    }
    int K1;
    // Виклик методу ps
    ps(A, out K1);
    // Виведення зміненого масиву в поле списку listBox2
    for (int i = 0; i < n; i++)
    {
        listBox2.Items.Add($"{A[i]}");
    }
    label4.Text = $"Кількість непарних чисел = {K1}";
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 6.6.

Рисунок 6.6 – Результати розв’язання задачі 6.2

Задача 6.3. Обчислити значення виразу

$$F = 1,5 \sqrt[3]{\sum_{i=1}^7 (x_i + a)} - 8,2 \sqrt{\sum_{i=0}^{10} (y_i + b)},$$

де $a = 3; b = 7$;

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

y_i – додатні елементи одновимірного масиву $Y(12)$; $i = 0, 1, \dots, 11$.

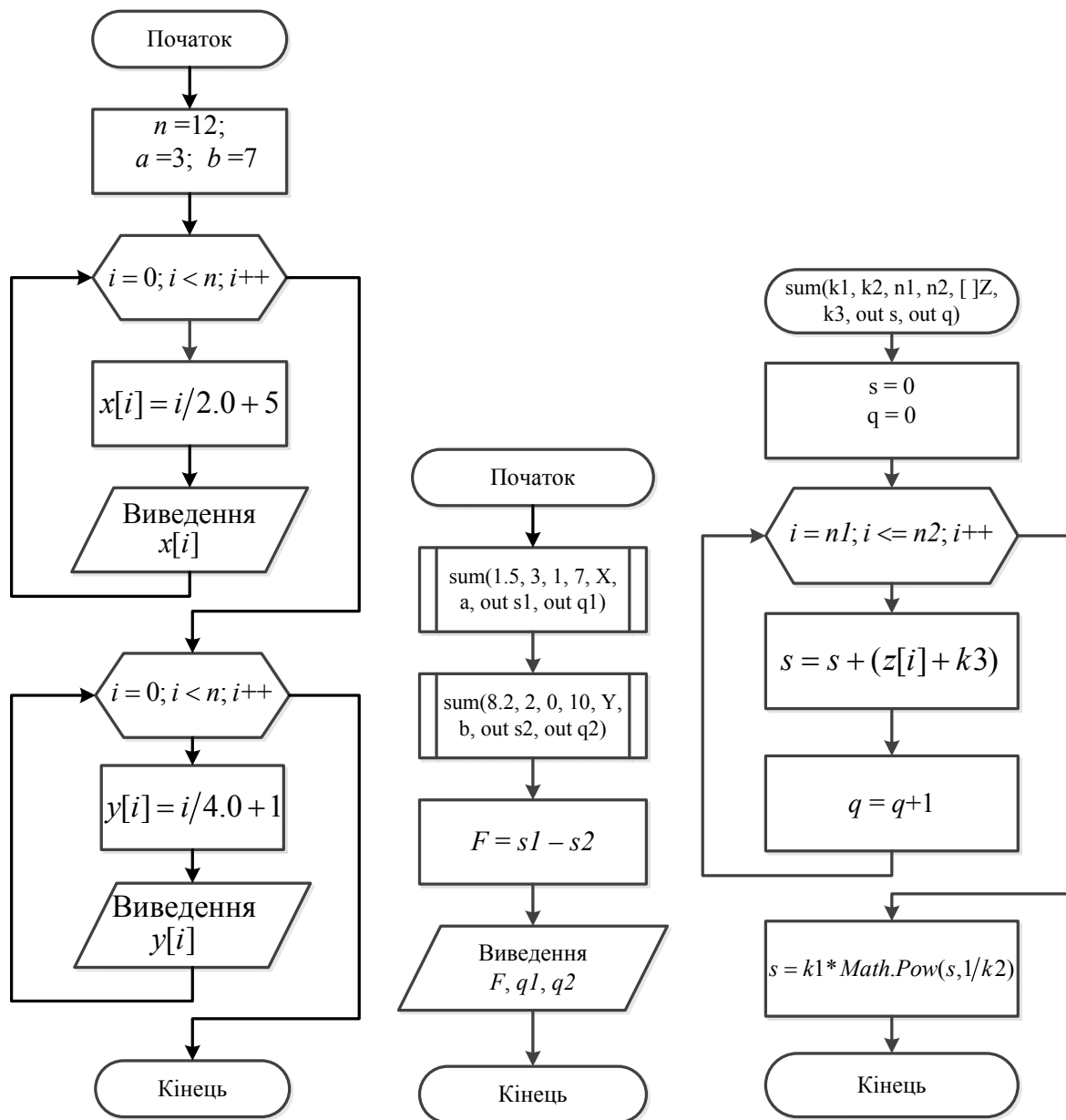
Елементи масивів обчислити за формулами:

$$x_i = i/2 + 5 ; \quad y_i = i/4 + 1.$$

У вигляді методу оформити обчислення суми $k1 \cdot k2 \sqrt{\sum_{i=n1}^{n2} (z_i + k3)}$ та розрахунок кількості елементів, що додаються.

Розробити *Windows-додаток*.

Блок-схему розв’язання задачі подано на рис. 6.7.



Оброблювачі подій cmdMas_Click, cmdF_Click і метод Sum()

Рисунок 6.7 – Блок-схема алгоритму до задачі 6.3

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти (рис. 6.8, 6.9).

На формі розташувати:

- три мітки (Label),
- два поля списку (ListBox),
- три командні кнопки (Button).



Рисунок 6.8 – Форма до задачі 6.3

2. Установіть значення властивостей об'єктів відповідно до табл. 6.3.

Таблиця 6.3 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
1	2	3	4
Форма	Form1	Name	Form1
Мітка	Label1	text	Масив X
Мітка	Label2	text	Масив Y
Мітка	Label3		
Поле списку	ListBox1		
Поле списку	ListBox2		
Командна кнопка	Button1	Name	cmdMas
		text	Обчислення і виведення елементів масивів
Командна кнопка	Button2	Name	cmdF
		text	Обчислення F
Командна кнопка	Button3	Name	cmdEnd
		text	Завершення роботи

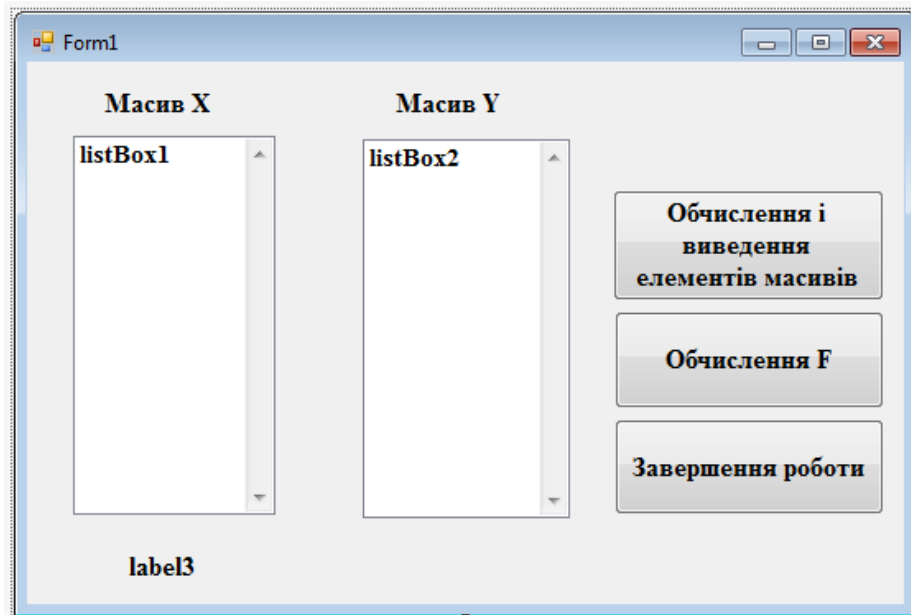


Рисунок 6.9 – Форма до задачі 6.3

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислення і виведення елементів масивів>, *клацання мишею по командній кнопці* <Обчислення F>, *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA16
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            int n; // Кількість елементів масиву
            double[] X;
            double[] Y;
            double a, b;
```

```

        static void sum(double k1, double k2, int n1, int n2,
double[] Z, double k3, out double s, out int q)
        {
            int i;
            s = 0; q = 0;
            for (i = n1; i <= n2; i++)
            {
                s = s + (Z[i] + k3);
                q = q + 1;
            }
            s = k1 * Math.Pow(s, (1 / k2));
        }

private void cmdMas_Click(object sender, EventArgs e)
{
    n = 12;
    X = new double[n];
    Y = new double[n];
    a = 3.0; b = 7.0;
    int i;
    for (i = 0; i < n; i++)
    {
        X[i] = i / 2.0 + 5;
        listBox1.Items.Add($"{X[i]:F2}");
    }
    for (i = 0; i < n; i++)
    {
        Y[i] = i / 4.0 + 1;
        listBox2.Items.Add($"{Y[i]:F2}");
    }
}

private void cmdF_Click(object sender, EventArgs e)
{
    double F, s1, s2;
    int q1, q2;
    sum(1.5, 3, 1, 7, X, a, out s1, out q1);
    sum(8.2, 2, 0, 10, Y, b, out s2, out q2);
    F = s1 - s2;
    label3.Text = $"F= {F:F3}   q1= {q1}   q2= {q2}";
}

```

```

        private void cmdEnd_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}

```

Результати розрахунку подано на рис. 6.10.

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two list boxes. The first list box, labeled "Масив X", contains the following values: 5,00, 5,50, 6,00, 6,50, 7,00, 7,50, 8,00, 8,50, 9,00, 9,50, 10,00, and 10,50. The second list box, labeled "Масив Y", contains the following values: 1,00, 1,25, 1,50, 1,75, 2,00, 2,25, 2,50, 2,75, 3,00, 3,25, 3,50, and 3,75. Below these list boxes, the text "F=-76,532 q1=7 q2=11" is displayed. To the right of the list boxes, there are three buttons: "Обчислення і виведення елементів масивів", "Обчислення F", and "Завершення роботи".

Рисунок 6.10 – Результати розв’язання задачі 6.3

Пояснення:

```

int n; // Кількість елементів масиву
double[] X;
double[] Y;
double a, b;

```

Величини, що оголошуються перед усіма методами, сприймаються як глобальні, тобто загальні й доступні стосовно всіх методів класу.

Тема 7. РЯДКИ

Мета роботи: вивчити правила роботи зі списком `ListBox`. Написати програму для роботи з рядками.

Задача 7.1. Написати програму підрахування кількості слів у довільному рядку. Як розділювач використовується пробіл. Для введення рядків використовувати список `ListBox`. Рядки уводяться на етапі проектування форми, використовуючи вікно властивостей. Виведення результатів розрахунку організувати на мітку `Label`.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 7.1, 7.2.

На формі розташувати:

- один список (`ListBox`),
- одну мітку (`Label`),
- дві командні кнопки (`Button`).

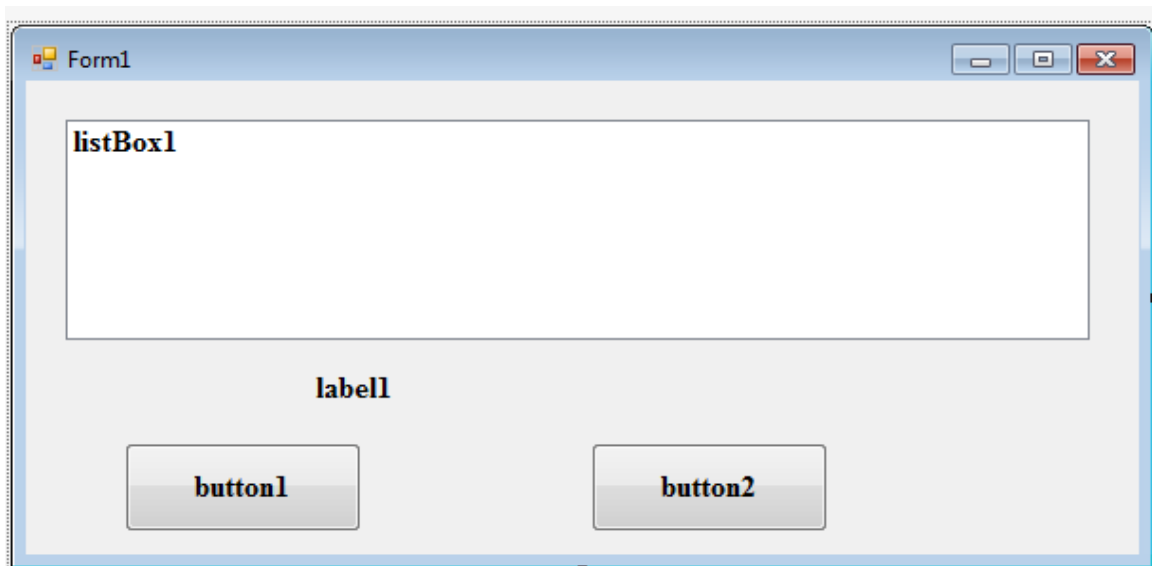


Рисунок 7.1 – Форма до задачі 7.1

2. Установіть значення властивостей об'єктів відповідно до табл. 7.1.

Таблиця 7.1– Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивостей
1	2	3	4
Форма	Form1		
Мітка	Label1	TextAlign	TopLeft
		Font	Times New Roman, Regular, 12
Список	ListBox1	Items	Задається весь необхідний список. Для цього слід вибрати властивість Items і натиснути розташовану праворуч кнопку із трьома точками. Відкриється діалогове вікно String Collection Editor (Редактор елементів списку). Після задання всіх елементів потрібно натиснути кнопку ОК
		SelectionMode	MultiSimple Клацання мишею на елементі виділяє його або знімає виділення
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
		Font	Times New Roman, Bold, 12
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи
		Font	Times New Roman, Bold, 12

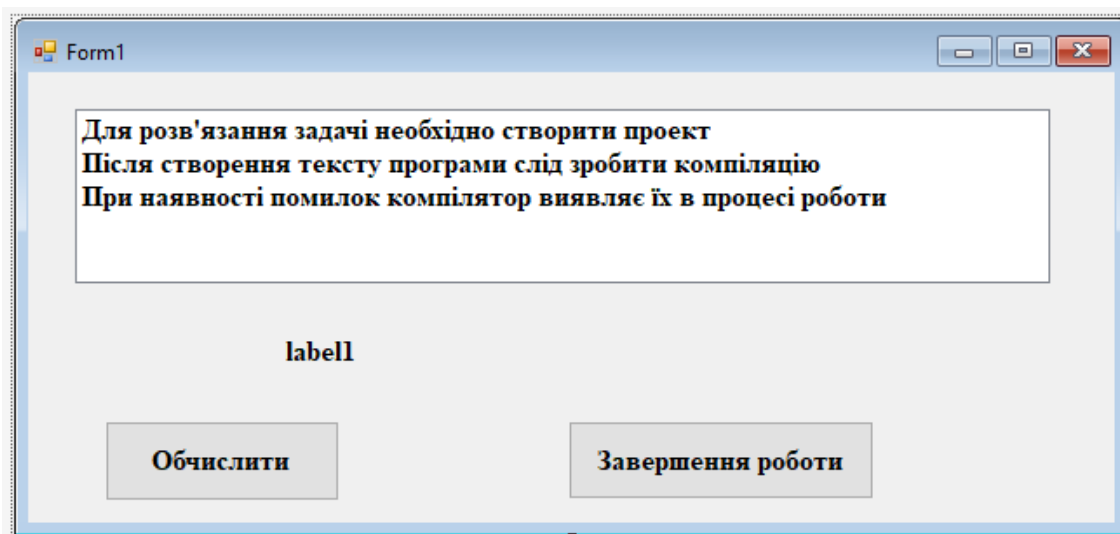


Рисунок 7.2 – Форма до задачі 7.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace WA25
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            // Одержуємо номер виділеного рядка
            int index = listBox1.SelectedIndex;
            // Зчитуємо рядок у змінну str
            string str = (string)listBox1.Items[index];
            // Дізнаємося про кількість символів у рядку
            int len = str.Length;
            // Задаємо початкову кількість пробілів, рівну 0
            int k = 0;
            int k1; //Кількість слів
            // Установлюємо лічильник символів у 0
            int i = 0;
            // Організуємо цикл перебору всіх символів у рядку
            while (i < len)
            {
                // Якщо знайшли пробіл, то
                // збільшуємо лічильник пробілів на 1
                if (str[i] == ' ')
                {
                    k = k + 1;
                    i = i + 1;
                }
                k1 = k + 1;
                label1.Text = $" Кількість слів = {k1}";
            }
        }
    }
}
```

```

        private void cmdEnd_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}

```

Результати розрахунку подано на рис. 7.3.

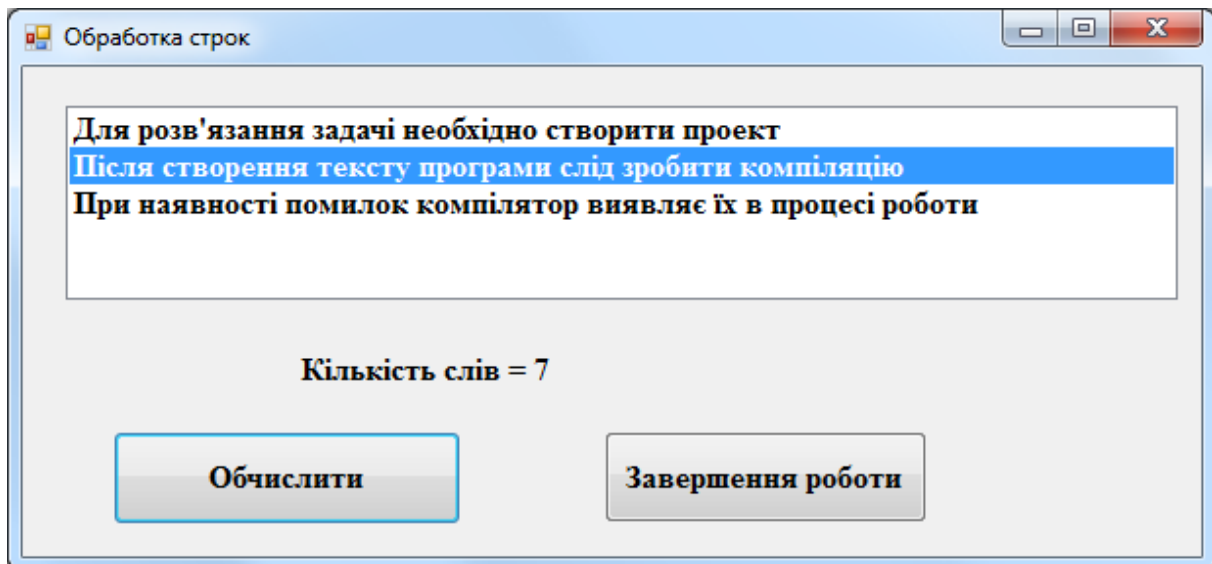
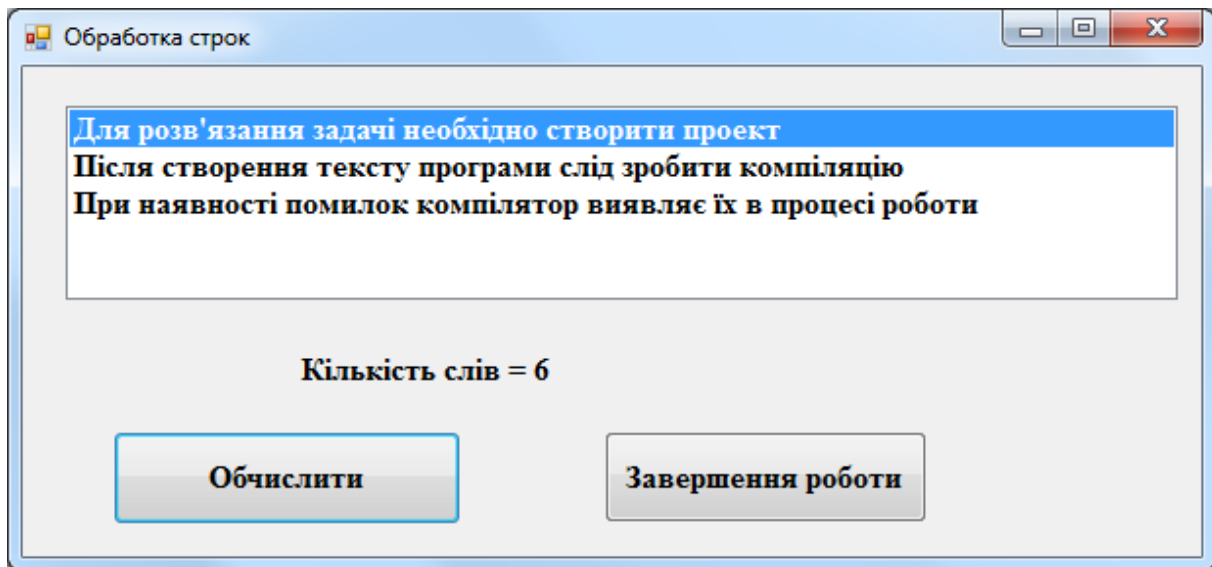


Рисунок 7.3 – Результати розв'язання задачі 7.1

Пояснення:

Елемент керування `ListBox` являє собою список, елементи якого вибираються за допомогою клавіатури або миші. Користувач може вибрати одне (властивість `SelectionMode` дорівнює `One`) або кілька значень (властивість `SelectionMode` дорівнює `MultiSimple` або `MultiExtended`). Якщо значення властивості `SelectionMode` дорівнює `MultiSimple`, клацання мишею на елементі виділяє його або знімає виділення. Значення `MultiExtended` дозволяє використовувати при виділенні діапазону рядків клавішу `Shift`, а при додаванні елемента – клавішу `Ctrl`, аналогічно провіднику Windows. Заборонити виділення можна, установивши значення властивості `SelectionMode`, рівне `None`.

Елементи списку нумеруються з нуля. Вони зберігаються у властивості `Items`, що являє собою колекцію. В `Items` можна додавати елементи за допомогою методів `Add`, `AddRange` і `Insert`. Для видалення елементів служать методи `Insert` і `RemoveAt`, що видаляють заданий елемент і елемент за заданим індексом відповідно.

Виділені елементи можна одержати за допомогою властивостей `SelectedItems` і `SelectedIndices`, що надають доступ до колекцій виділених елементів і їх індексів.

Об'єкт `Items` зберігає об'єкти, що перебувають у списку. Об'єкт може бути будь-яким класом – дані класу перетворюються для відображення в рядкове подання методом `ToString()`. У цій задачі як об'єкт виступають рядки. Однак оскільки об'єкт `Items` зберігає об'єкти, наведені до типу `Object`, перед використанням необхідно привести їх назад до початкового типу, у цій задачі – до типу `string`

```
string a= (string)listBox1.Items[0];
```

Для визначення номера виділеного елемента використовується властивість `SelectedIndex`.

Тема 8. ФАЙЛИ

Задача 8.1. Записати у файл значення функції

$$y(x) = \cos(x), \quad -3 \leq x \leq 3.$$

Прочитати створений файл. Створити Windows-додаток.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 8.1, 8.2.

На формі розташуйте: одне поле списку (ListBox); дві командні кнопки (Button).

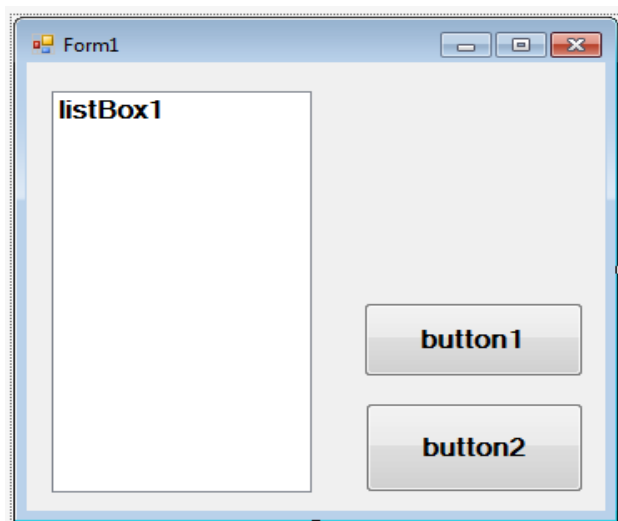


Рисунок 8.1 – Форма до задачі 8.1

2. Установіть значення властивостей об'єктів відповідно до табл. 8.1.

Таблиця 8.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1		
Поле списку	ListBox1		
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи

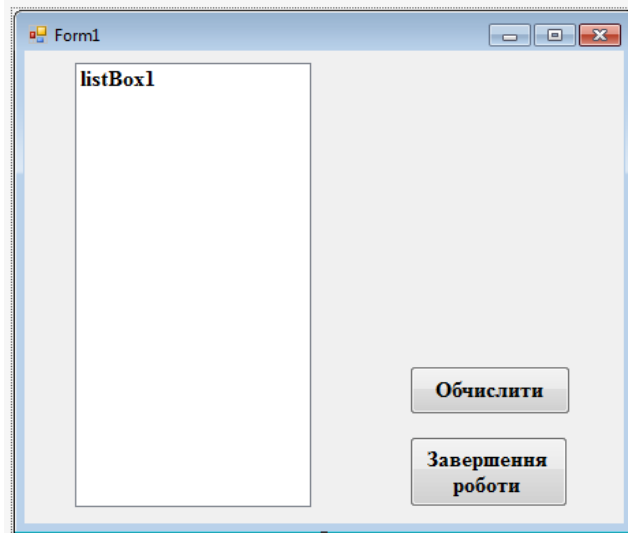


Рисунок 8.2 – Форма до задачі 8.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;
using System.IO;

namespace WA18
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            // Файл відкривається для запису
            FileStream fsW = new FileStream("D:\\file7.txt",
                                           FileMode.Create, FileAccess.Write);
            StreamWriter sw = new StreamWriter(fsW);
            double x, y;    // аргумент і значення функції
            string s;
            // Запис до файлу назви стовпців
            sw.WriteLine("-----");
            sw.WriteLine("    x            y    ");
            sw.WriteLine("-----");
        }
    }
}
```

```

for (x = -3; x <= 3; x = x + 0.5)
{
    y = 5 * x + 3 * Math.Sin(x) + 7;
    // Обчислені значення у записуються до файлу
    s = $"{x,7:F2} {y,7:F3}";
    sw.WriteLine(s);
}
sw.Close();
fsW.Close();
// Файл відкривається для читання
FileStream fsR = new FileStream("D:\\file7.txt",
                                FileMode.Open, FileAccess.Read);
StreamReader sr = new StreamReader(fsR);
// Цикл виконується, доки не буде досягнутий кінець файлу
while ((s = sr.ReadLine()) != null)
    listBox1.Items.Add(s);
sr.Close();
fsR.Close();
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 8.3.

x	y
-3,00	-8,423
-2,50	-7,295
-2,00	-5,728
-1,50	-3,492
-1,00	-0,524
-0,50	3,062
0,00	7,000
0,50	10,938
1,00	14,524
1,50	17,492
2,00	19,728
2,50	21,295
3,00	22,423

Обчислити

Завершення роботи

Рисунок 8.3 – Результати розв’язання задачі 8.1

Тема 9. ГРАФІКИ ФУНКЦІЙ

Мета роботи: вивчити можливості побудови графіків за допомогою елемента керування Chart. Написати програму побудови на екрані графіка заданої функції.

Задача 9.1. Скласти програму, що відображає графіки функцій $\sin(x)$ і $\cos(x)$ на інтервалі $[X_{\min}, X_{\max}]$.

ПОРЯДОК ДІЙ

1. Розташуйте на формі об'єкти, як показано на рис. 9.1, 9.2.

На формі розташувати:

- елемент керування Chart;
- три мітки (Label);
- три текстових поля (TextBox);
- дві командні кнопки (Button).

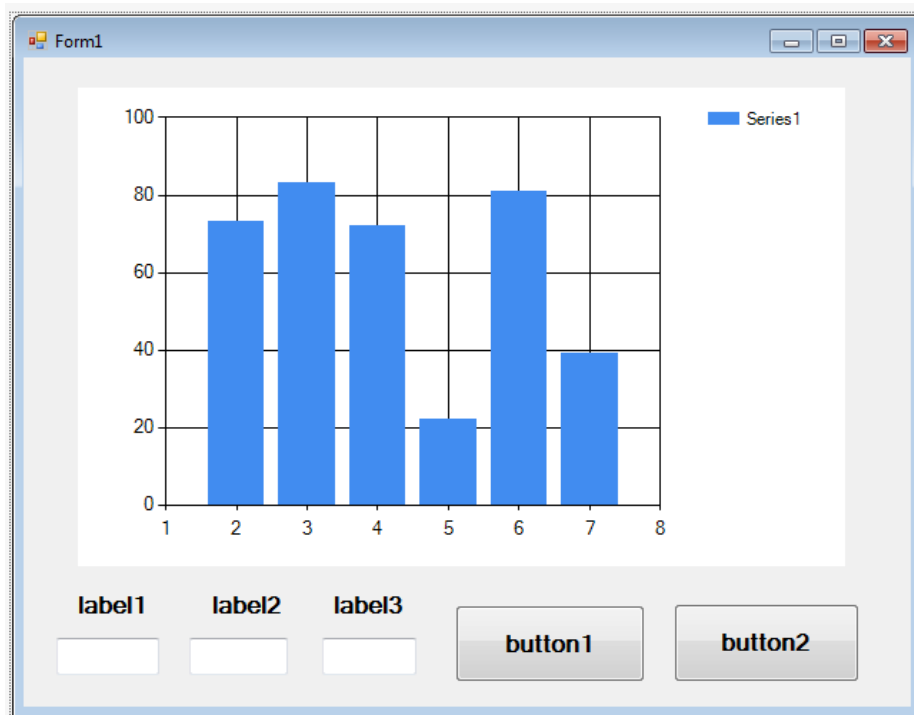


Рисунок 9.1 – Форма до задачі 9.1

2. Установіть значення властивостей об'єктів відповідно до табл. 9.1.

Таблиця 9.1 – Значення властивостей об'єктів

Об'єкт	Ім'я об'єкта за замовчуванням (значення властивості Name)	Властивість	Значення властивості
Форма	Form1	Text	Побудова графіка функції
Мітка	Label1	Text	Xmin
Мітка	Label2	Text	Xmax
Мітка	Label3	Text	крок
Текстове поле	TextBox1	Name	txtXmin
Текстове поле	TextBox2	Name	txtXmax
Текстове поле	TextBox3	Name	txtStep
Елемент керування Chart (розташовується в панелі елементів у розділі Дані)	chart1	Series	<p>Список графіків зберігається у властивості Series, який можна змінити, вибравши відповідний пункт у вікні властивостей.</p> <p>1. Оскільки на одному полі потрібно вивести два окремі графіки функцій, потрібно додати ще один елемент.</p> <p>2. Обидва елементи потрібно настроїти: змінити тип діаграми ChartType на Spline. Тут же можна змінити підписи до графіків з абстрактних Series1 і Series2 на $\sin(x)$ і $\cos(x)$ – за це відповідає властивість LegendText.</p> <p>3. За допомогою властивості Bordewidth можна зробити лінію графіка товстішою, а потім поміняти колір лінії за допомогою властивості Color</p>
Командна кнопка	Button1	Name	cmdStart
		Text	Обчислити
Командна кнопка	Button2	Name	cmdEnd
		Text	Завершення роботи

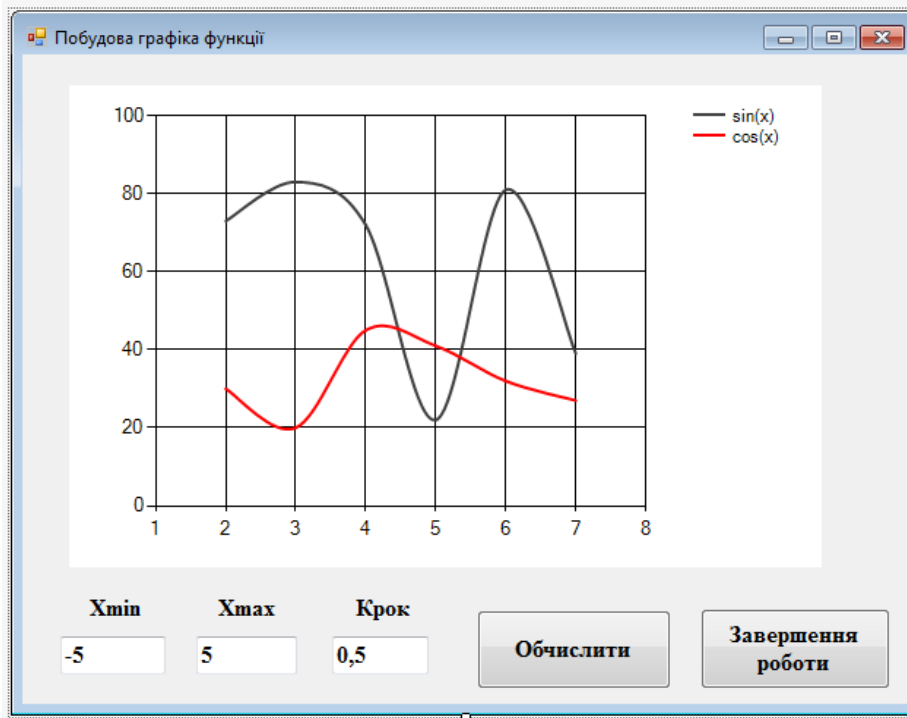


Рисунок 9.2 – Форма до задачі 9.1

3. Програмування

Напишіть оброблювачі подій *клацання мишею по командній кнопці* <Обчислити> і *клацання мишею по командній кнопці* <Завершення роботи>.

Програмний код:

```
using System;
using System.Windows.Forms;

namespace W30
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdStart_Click(object sender, EventArgs e)
        {
            // Зчитуємо з текстових полів необхідні значення
            double Xmin = Convert.ToDouble(txtXmin.Text);
            double Xmax = Convert.ToDouble(txtXmax.Text);
        }
    }
}
```

```

double Step = Convert.ToDouble(txtStep.Text);
// Кількість точок графіка
int count = (int)Math.Ceiling((Xmax - Xmin) / Step) + 1;
// Масив значень x - загальний для обох графіків
double[] x = new double[count];
// Оголошення двох масивів y1 і y2
// для побудови графіків
double[] y1 = new double[count];
double[] y2 = new double[count];
// Розраховуємо точки для графіків функції
for (int i = 0; i < count; i++)
{
    // Обчислюємо значення x
    x[i] = Xmin + Step * i;
    // Обчислюємо значення функції в точці x;
    y1[i] = Math.Sin(x[i]);
    y2[i] = Math.Cos(x[i]);
}

// Настроюємо осі графіка
chart1.ChartAreas[0].AxisX.Minimum = Xmin;
chart1.ChartAreas[0].AxisX.Maximum = Xmax;

// визначаємо крок сітки
chart1.ChartAreas[0].AxisX.MajorGrid.Interval = Step;

// Додаємо обчислені значення в графіки
chart1.Series[0].Points.DataBindXY(x, y1);
chart1.Series[1].Points.DataBindXY(x, y2);
}

private void cmdEnd_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

Результати розрахунку подано на рис. 9.3.

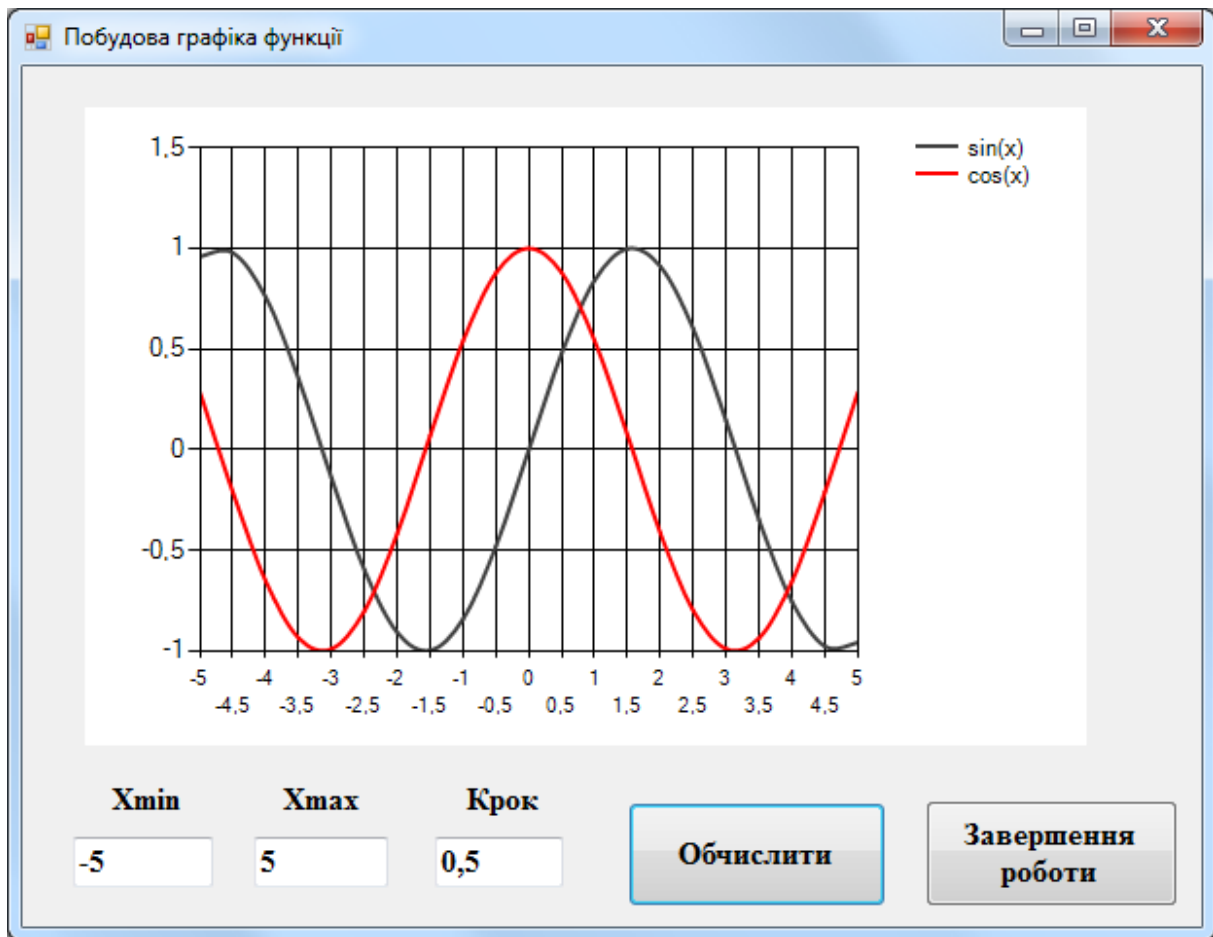


Рисунок 9.3 – Результати розв’язання задачі 9.1

Пояснення:

Побудова графіка (діаграми) проводиться після обчислення таблиці значень функції $y = f(x)$ на інтервалі $[Xmin, Xmax]$ із заданим кроком. Отримана таблиця передається в спеціальний масив **Points** об’єкта **Series** елемента керування **Chart** за допомогою методу **DataBindXY**.

Елемент керування **Chart** здійснює всю роботу з відображення графіків: будує й розмічає осі, рисує координатну сітку, підписує назви осей і самого графіка, відображає передану таблицю у вигляді всіляких графіків або діаграм. В елементі керування **Chart** можна настроїти товщину, стиль і колір ліній, параметри шрифту підписів, кроки розмітки координатної сітки й багато чого іншого. У процесі роботи програми зміна параметрів можлива через звертання до відповідних властивостей елемента керування **Chart**. Так, наприклад, властивість **AxisX** містить значення максимальної межі нижньої осі графіка, і при його зміні під час роботи програми автоматично змінюється зображення графіка.