**Лабораторна робота №1.** **Використання базових типів, консольного введення / виведення та операторів мови C#.**

**Виконавець: Лазоренко М.Д.**

**Група виконавця: КН-922в**

**Мета роботи:** набути навички в написанні програм з використання базових типів, операторів та засобів введення / виведення.

**Індивідуальні завдання**

**Варіант №9**

**Завдання 1**

Використовуючи завдання власного варіанта до лабораторної роботи 2 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 2:* *Розробити програму для введення фактичних даних з таблиці, представленої у виданому варіанті індивідуальних завдань, і для виведення на екран таблиці, подібної тій, яка знаходиться в індивідуальному завданні (включаючи заголовок і примітки).*



***Текст програми:***

using System;

class task\_01

{

static void Main()

{

string name1, name2, name3;

string producer1, producer2, producer3;

ushort components1, components2, components3;

float price1, price2, price3;

/\* Введення фактичних даних \*/

Console.WriteLine("1. Введіть: назву, виробника, кількість складових частин, ціну > ");

var input1 = Console.ReadLine().Split(' ');

name1 = input1[0];

producer1 = input1[1];

components1 = ushort.Parse(input1[2]);

price1 = float.Parse(input1[3]);

Console.WriteLine("2. Введіть: назву, виробника, кількість складових частин, ціну > ");

var input2 = Console.ReadLine().Split(' ');

name2 = input2[0];

producer2 = input2[1];

components2 = ushort.Parse(input2[2]);

price2 = float.Parse(input2[3]);

Console.WriteLine("3. Введіть: назву, виробника, кількість складових частин, ціну > ");

var input3 = Console.ReadLine().Split(' ');

name3 = input3[0];

producer3 = input3[1];

components3 = ushort.Parse(input3[2]);

price3 = float.Parse(input3[3]);

/\* Виведення таблиці \*/

Console.WriteLine("----------------------------------------------------------");

Console.WriteLine("| Офісні пакети |");

Console.WriteLine("|--------------------------------------------------------|");

Console.WriteLine("| Назва | Виробник | К-сть складових | Ціна ($) |");

Console.WriteLine("|------------|------------|------------------|-----------|");

/\* Виведення фактичних даних \*/

Console.WriteLine($"| {name1,10} | {producer1,10} | {components1,2} | {price1,7:0.00} |");

Console.WriteLine($"| {name2,10} | {producer2,10} | {components2,2} | {price2,7:0.00} |");

Console.WriteLine($"| {name3,10} | {producer3,10} | {components3,2} | {price3,7:0.00} |");

/\* Виведення примітки \*/

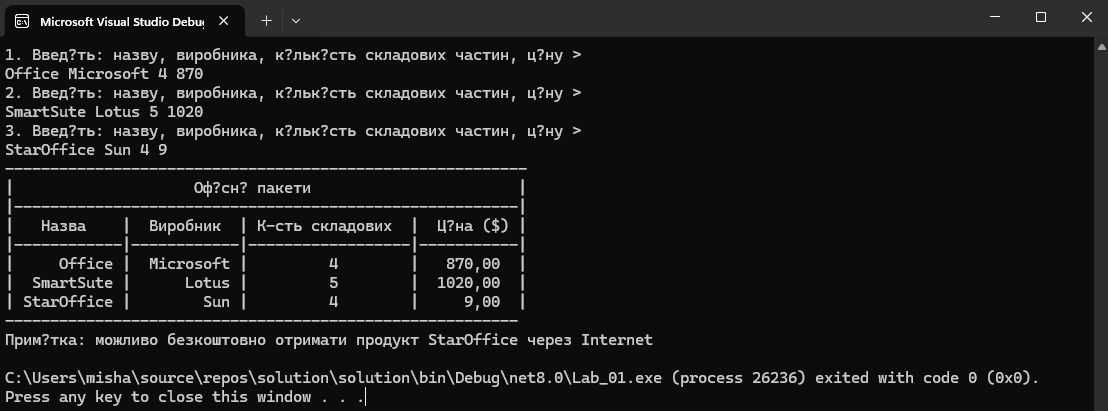
Console.WriteLine("---------------------------------------------------------");

Console.WriteLine("Примітка: можливо безкоштовно отримати продукт StarOffice через Internet");

}

}

***Результат роботи програми:***



***Опис програми:***

Ця програма на мові C# призначена для введення та виведення інформації про три різні офісні пакети. Вона запитує користувача про назву, виробника, кількість складових частин і ціну кожного з трьох пакетів, а потім виводить введені дані у вигляді таблиці на екран.

Програма дозволяє користувачу вводити фактичні дані для кожного пакета, використовуючи консольний інтерфейс, а також форматовано виводить результати у вигляді таблиці.

***Основні функції програми:***

1. *Введення даних:* Користувач вводить назву, виробника, кількість складових частин та ціну для кожного з трьох офісних пакетів.
2. *Виведення таблиці:* Програма будує таблицю, яка відображає інформацію у форматі:

* Назва пакета
* Виробник
* Кількість складових частин
* Ціна в доларах США

1. *Примітка:* В кінці програма додає примітку, що продукт StarOffice можна отримати безкоштовно через Інтернет.

***Ієрархія та структура класів:***

У даній програмі використовується лише один клас task\_01 з одним методом Main, який є точкою входу в програму.

**Клас task\_01**

* **Метод Main:**
  + Запитує введення даних для трьох офісних пакетів через консоль.
  + Зберігає введені дані в змінні.
  + Форматує та виводить ці дані у вигляді таблиці.
  + Виводить примітку про продукт StarOffice.

**Завдання 2\***

Виконати минуле завдання, покращивши зовнішній вигляд таблиці із вихідними результатами за допомогою символів псевдографіки (┐, ┌, ┘, └, ├, ┤, ┬, ┴ та ін.) та налаштування кольорів консолі, наприклад, числа відобразити одним кольором, текст – іншим, таблицю – іншим від попередніх.

***Текст програми:***

using System;

class task\_02

{

static void Main()

{

// Налаштування кольорів для консолі

Console.ForegroundColor = ConsoleColor.Yellow; // Колір для таблиці

// Введення фактичних даних

string name1, name2, name3;

string producer1, producer2, producer3;

ushort components1, components2, components3;

float price1, price2, price3;

Console.ResetColor();

Console.ForegroundColor = ConsoleColor.Green; // Колір для тексту

Console.WriteLine("1. Введіть: назву, виробника, кількість складових частин, ціну > ");

var input1 = Console.ReadLine().Split(' ');

name1 = input1[0];

producer1 = input1[1];

components1 = ushort.Parse(input1[2]);

price1 = float.Parse(input1[3]);

Console.WriteLine("2. Введіть: назву, виробника, кількість складових частин, ціну > ");

var input2 = Console.ReadLine().Split(' ');

name2 = input2[0];

producer2 = input2[1];

components2 = ushort.Parse(input2[2]);

price2 = float.Parse(input2[3]);

Console.WriteLine("3. Введіть: назву, виробника, кількість складових частин, ціну > ");

var input3 = Console.ReadLine().Split(' ');

name3 = input3[0];

producer3 = input3[1];

components3 = ushort.Parse(input3[2]);

price3 = float.Parse(input3[3]);

Console.ResetColor();

Console.ForegroundColor = ConsoleColor.Yellow; // Повернення кольору таблиці

// Верх таблиці

Console.WriteLine("┌────────────┬────────────┬─────────────────┬─────────────┐");

Console.WriteLine("│ Офісні пакети │");

Console.WriteLine("├────────────┼────────────┼─────────────────┼─────────────┤");

Console.WriteLine("│ Назва │ Виробник │ К-сть складових │ Ціна ($) │");

Console.WriteLine("├────────────┼────────────┼─────────────────┼─────────────┤");

// Виведення фактичних даних з використанням псевдографіки

Console.WriteLine($"│ {name1,10} │ {producer1,10} │ {components1,2}│ {price1,7:0.00}│");

Console.WriteLine($"│ {name2,10} │ {producer2,10} │ {components2,2}│ {price2,7:0.00}│");

Console.WriteLine($"│ {name3,10} │ {producer3,10} │ {components3,2}│ {price3,7:0.00}│");

// Нижня частина таблиці

Console.WriteLine("├────────────┴────────────┴─────────────────┴─────────────┤");

Console.WriteLine("│ Примітка: можливо безкоштовно отримати продукт │");

Console.WriteLine("│ StarOffice через Internet │");

Console.WriteLine("└─────────────────────────────────────────────────────────┘");

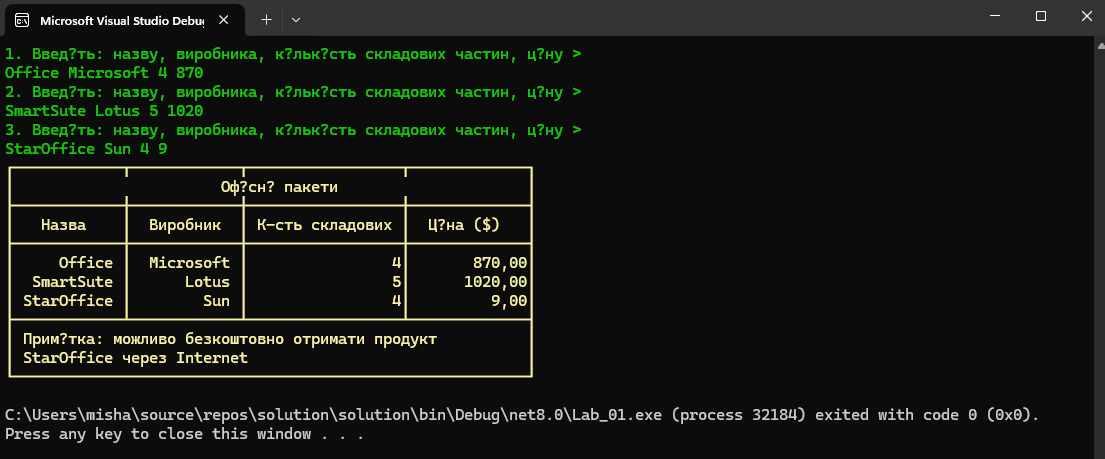
// Повернення кольору за замовчуванням

Console.ResetColor();

}

}

***Результат роботи програми:***



***Опис програми:***

Програма на мові C# реалізує консольний додаток для введення та відображення інформації про три офісні пакети. Під час виконання користувач вводить назву, виробника, кількість складових частин та ціну кожного пакета, після чого ці дані форматовано виводяться у вигляді таблиці з використанням псевдографіки та кольорового відображення.

Програма також використовує можливість зміни кольорів у консолі для поліпшення візуального відображення, зокрема:

* Жовтий колір для таблиці,
* Зелений колір для тексту введення.

***Основні функції програми:***

1. **Налаштування кольорів консолі**: Програма змінює колір тексту в залежності від контексту (введення даних, виведення таблиці).
2. **Введення даних**: Користувач вводить інформацію про кожен з трьох офісних пакетів через консоль.
3. **Виведення таблиці**: Програма виводить результати у вигляді таблиці з використанням псевдографіки, що робить таблицю більш структурованою.
4. **Примітка**: В кінці таблиці виводиться примітка про можливість безкоштовного отримання продукту StarOffice через Інтернет.

***Ієрархія та структура класів:***

У програмі використовується один клас task\_02, який містить лише метод Main. Цей метод виконує всі необхідні дії — від введення даних до їх форматованого виведення.

**Клас task\_02**

* **Метод Main**:
  + Встановлює кольори для консолі перед введенням та виведенням інформації.
  + Запитує у користувача введення назви, виробника, кількості складових частин та ціни для кожного з трьох пакетів.
  + Форматує введені дані у вигляді таблиці з псевдографікою.
  + Повертає колір консолі до початкового стану після виконання всіх дій.

**Завдання 3**

Використовуючи завдання власного варіанта до лабораторної роботи 3 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 3: Складіть програму, яка підраховує і виводить на екран значення змінних* ***t1*** *і* ***t2,*** *що обчислюються за формулами, які наведені у виданому варіанті індивідуального завдання. Визначте області допустимих значень змінних, які присутні в формулах, і задайте довільні значення з цих областей. Змінні, які мають імена* ***m*** *і* ***n*** *– цілі, решта змінних мають тип – з рухомою крапкою. Значення змінних з іменами* ***x*** *і* ***у*** *повинні вводитися з клавіатури, значення інших – задаватися як початкові значення при оголошенні відповідних змінних. Допускається (і навіть бажано) спростити/розкласти формули для того, щоб забезпечити мінімізацію об’єму обчислень.*

**Варіант № 9**





***Текст програми:***

using System;

class task\_03

{

static void Main()

{

// Початкові значення змінних

double a = 2.0;

double b = 3.0;

// Введення значень x і y з клавіатури

Console.Write("Введіть значення x: ");

double x = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть значення y: ");

double y = Convert.ToDouble(Console.ReadLine());

// Обчислення t1 і t2

double t1 = (1 / Math.Pow(b, 2)) \* (Math.Pow(Math.Log(x), 2) + (a \* x / y));

double t2 = (x / a) - Math.Tan(a \* x / 2) + (2 / Math.Pow(a, 2)) \* Math.Log(Math.Sin(a \* x / 2));

// Виведення результатів

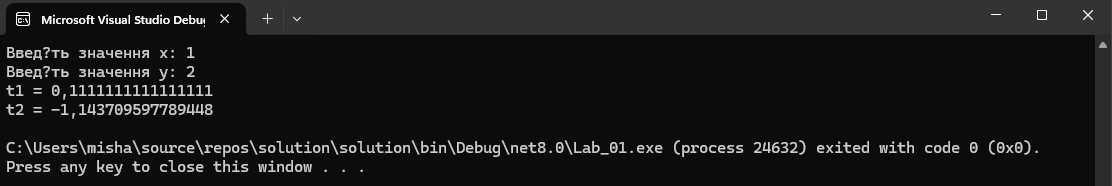
Console.WriteLine($"t1 = {t1}");

Console.WriteLine($"t2 = {t2}");

}

}

***Результат роботи програми:***



***Опис програми:***

Програма на мові C# виконує обчислення двох математичних виразів, використовуючи значення, введені користувачем. Користувач вводить значення змінних x і y, після чого програма обчислює два вирази t1 та t2 на основі цих значень, а також попередньо визначених змінних a і b. Обчислені результати виводяться на екран.

***Основні функції програми:***

1. Ініціалізація початкових значень для змінних a і b.

2. Введення значень x та y з клавіатури.

3. Обчислення двох виразів:

* t1 = (1 / b²) \* (ln(x)² + (a \* x / y))
* t2 = (x / a) - tan(a \* x / 2) + (2 / a²) \* ln(sin(a \* x / 2))

4. Виведення обчислених результатів на екран.

***Ієрархія та структура класів:***

У програмі використовується один клас task\_03, що містить основний метод Main, де відбувається весь процес обчислення та взаємодії з користувачем.

**Клас task\_03**

* **Метод Main**:
  + Ініціалізує значення змінних a = 2.0 та b = 3.0.
  + Використовує метод Console.ReadLine() для зчитування значень x і y, які вводить користувач.
  + Використовує математичні функції Math.Pow(), Math.Log(), Math.Tan(), і Math.Sin() для обчислення виразів t1 та t2.
  + Виводить результати обчислень на екран за допомогою Console.WriteLine().

**Завдання 4\***

Виконати минуле завдання, покращивши гнучкість програми обчислення виразу, якщо вхідні дані є некоректними із точки зору математичних правил (нуль в знаменнику, від’ємне значення під знаком квадратного кореня та ін.). Для цього використати механізм обробки відповідних виключень (try … catch … finally)

***Текст програми:***

using System;

class task\_04

{

static void Main()

{

// Початкові значення змінних

double a = 2.0;

double b = 3.0;

try

{

// Введення значень x і y з клавіатури

Console.Write("Введіть значення x: ");

double x = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть значення y: ");

double y = Convert.ToDouble(Console.ReadLine());

// Перевірка на некоректні значення

if (y == 0)

{

throw new DivideByZeroException("Значення y не може бути нулем.");

}

if (x <= 0)

{

throw new ArgumentException("Значення x повинно бути більше нуля для обчислення логарифму.");

}

// Обчислення t1 і t2

double t1 = (1 / Math.Pow(b, 2)) \* (Math.Pow(Math.Log(x), 2) + (a \* x / y));

double t2 = (x / a) - Math.Tan(a \* x / 2) + (2 / Math.Pow(a, 2)) \* Math.Log(Math.Sin(a \* x / 2));

// Виведення результатів

Console.WriteLine($"t1 = {t1}");

Console.WriteLine($"t2 = {t2}");

}

catch (DivideByZeroException ex)

{

Console.WriteLine($"Помилка: {ex.Message}");

}

catch (ArgumentException ex)

{

Console.WriteLine($"Помилка: {ex.Message}");

}

catch (Exception ex)

{

Console.WriteLine($"Невідома помилка: {ex.Message}");

}

finally

{

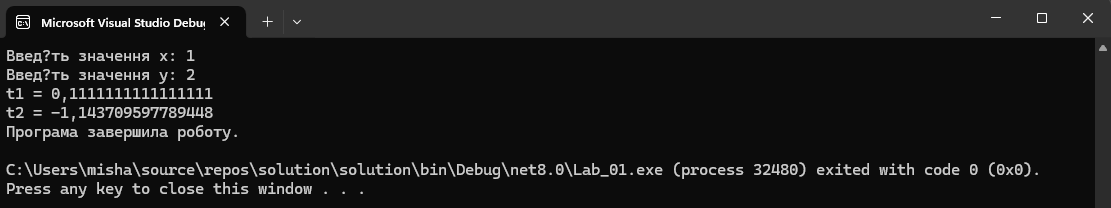
Console.WriteLine("Програма завершила роботу.");

}

}

}

***Результат роботи програми:***



***Опис програми:***

Програма на мові C# виконує обчислення двох математичних виразів з використанням значень, введених користувачем. Вона включає обробку виключень, щоб уникнути помилок при введенні некоректних даних, таких як нульове значення для змінної y або некоректне значення для логарифмічних обчислень.

***Основні функції програми:***

**1.** **Введення даних:** Користувач вводить значення для змінних x та y.

**2.** **Перевірка даних:** Програма перевіряє, чи є значення y нулем (що викликає помилку ділення на нуль) та чи є значення x більшим за нуль (що необхідно для обчислення логарифму).

**3.** **Обчислення математичних виразів:**

* t1 = (1 / b²) \* (ln(x)² + (a \* x / y))
* t2 = (x / a) - tan(a \* x / 2) + (2 / a²) \* ln(sin(a \* x / 2))

**4.** **Обробка виключень:** Якщо виникають помилки (наприклад, ділення на нуль або передача некоректного аргументу для логарифма), вони обробляються за допомогою try-catch блоку.

**5. Виведення результатів:** Результати обчислень або повідомлення про помилку виводяться на екран.

**6.** **Блок finally:** Програма завжди завершується виведенням повідомлення про завершення роботи, незалежно від того, чи виникла помилка.

***Ієрархія та структура класів:***

Програма складається з одного класу task\_04, який містить метод Main для виконання основної логіки.

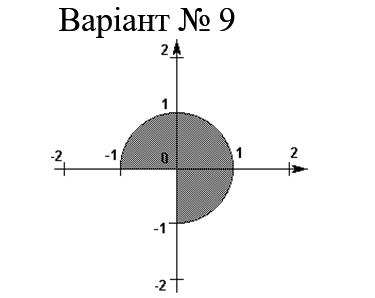
**Клас task\_04**

* **Метод Main**:
  + Ініціалізує початкові значення змінних a = 2.0 та b = 3.0.
  + Використовує try-catch-finally блок для обробки виключень, пов'язаних із некоректними даними:
    - DivideByZeroException для нульового значення змінної y.
    - ArgumentException для некоректного значення змінної x при обчисленні логарифму.
    - Загальний Exception для інших непередбачених помилок.
  + Обчислює математичні вирази t1 і t2, використовуючи бібліотечні функції для логарифмічних та тригонометричних операцій.
  + Виводить на екран результати або повідомлення про помилки.
  + Блок finally гарантує, що повідомлення про завершення програми буде виведено завжди.

**Завдання 5**

Використовуючи завдання власного варіанта до лабораторної роботи 5 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 5:* *Побудувати програму, яка вводить координати точки (****x****,* ***у****) і визначає, чи потрапляє точка в заштриховану область на малюнку, який відповідає виданому варіанту. Попадання на межу області вважати попаданням в область.*

****

***Текст програми:***

using System;

class task\_05

{

static void Main()

{

Console.WriteLine("Введіть координати точки (x, y):");

double x = Convert.ToDouble(Console.ReadLine());

double y = Convert.ToDouble(Console.ReadLine());

if (IsPointInShadedArea(x, y))

{

Console.WriteLine("Точка потрапляє в заштриховану область.");

}

else

{

Console.WriteLine("Точка не потрапляє в заштриховану область.");

}

}

static bool IsPointInShadedArea(double x, double y)

{

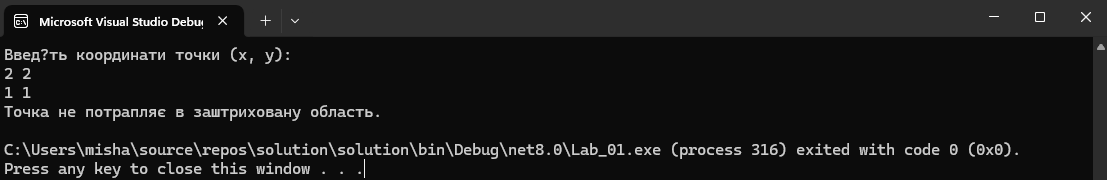
// Перевірка, чи точка знаходиться в верхній півколі з радіусом 1

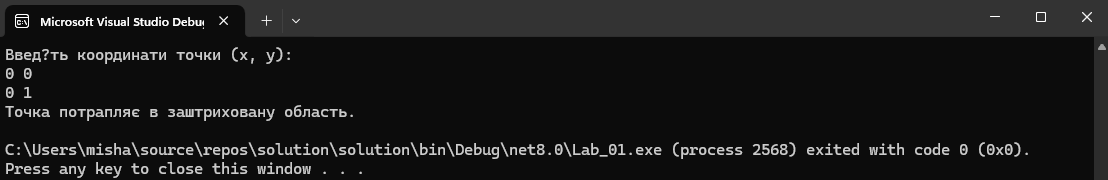
return (x \* x + y \* y <= 1) && (y >= 0);

}

}

***Результат роботи програми:***



******

***Опис програми:***

Ця програма на C# перевіряє, чи потрапляє задана точка з координатами (x, y) у заштриховану область на площині. Заштрихована область представляє собою верхню півкулю кола з радіусом 1, центр якого знаходиться на початку координат (0, 0).

***Основні функції програми:***

**1.** **Введення даних:** Програма запитує у користувача координати точки (x, y).

**2.** **Обробка даних:** Функція IsPointInShadedArea() перевіряє, чи точка потрапляє в верхню півкулю кола. Умова включає:

* Коло радіусом 1: перевірка, що відстань від точки до центру менша або дорівнює 1 (x² + y² ≤ 1).
* Верхня півкуля: перевірка, що y ≥ 0.

**3.** **Виведення результату:** Програма виводить повідомлення про те, чи потрапляє точка в заштриховану область.

***Ієрархія та структура класів:***

**Клас task\_05:** Головний клас програми, містить метод Main і статичний метод IsPointInShadedArea.

* **Метод Main:**
  + Відповідає за взаємодію з користувачем (отримання координат точки) та виведення результату.
  + Викликає функцію IsPointInShadedArea() для перевірки координат точки.
* **Метод IsPointInShadedArea:**
  + Статична функція, яка виконує математичну перевірку, чи потрапляє точка в заштриховану область.
  + Параметри: координати точки (x, y).
  + Повертає true, якщо точка знаходиться в заштрихованій області (верхня півкуля кола радіусом 1), і false в іншому випадку.

**Завдання 6\***Виконати минуле завдання без використання оператору if у будь-яких формах

***Текст програми:***

using System;

class task\_06

{

static void Main()

{

Console.WriteLine("Введіть координати точки (x, y):");

double x = Convert.ToDouble(Console.ReadLine());

double y = Convert.ToDouble(Console.ReadLine());

// Використовуємо тернарний оператор для визначення результату

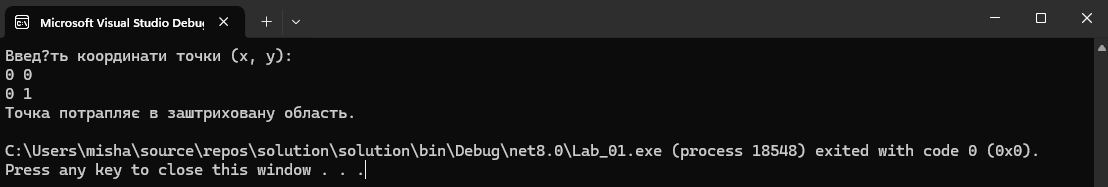
string result = (x \* x + y \* y <= 1) && (y >= 0) ? "Точка потрапляє в заштриховану область." : "Точка не потрапляє в заштриховану область.";

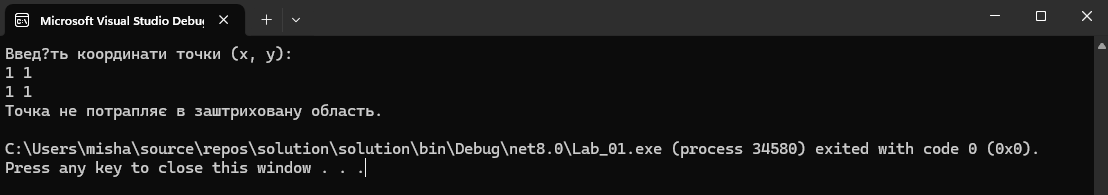
Console.WriteLine(result);

}

}

***Результат роботи програми:***

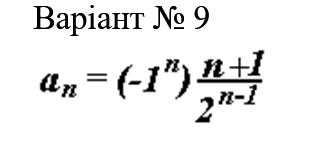


****

**Завдання 7**

Використовуючи завдання власного варіанта до лабораторної роботи 6 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 6:* *Для ряду, члени якого обчислюються за формулою, відповідною виданому індивідуальному завданню, знайти суму членів ряду з точністю до 0.000001 і суму перших 10 членів ряду. Якщо доцільно, можна спростити або перетворити вираз.*

******

***Текст програми:***

using System;

class task\_07

{

static void Main()

{

double epsilon = 0.000001;

double sum = 0.0;

double term;

int n = 0;

// Обчислення суми членів ряду з точністю до 0.000001

do

{

term = Math.Pow(-1, n) \* ((n + 1) / (Math.Pow(2, n-1)));

sum += term;

n++;

} while (Math.Abs(term) > epsilon);

Console.WriteLine("Сума членів ряду з точністю до 0.000001: " + sum);

// Обчислення суми перших 10 членів ряду

sum = 0.0;

for (n = 0; n < 10; n++)

{

term = Math.Pow(-1, n) \* ((n + 1) / (Math.Pow(2, n - 1)));

sum += term;

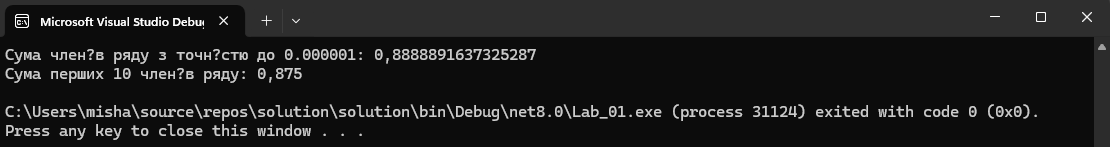
}

Console.WriteLine("Сума перших 10 членів ряду: " + sum);

}

}

***Результат роботи програми:***



***Опис програми:***

Ця програма на C# обчислює суму членів нескінченного ряду двома способами:

1. **Сума ряду з точністю до 0.000001:** Програма обчислює суму членів ряду до тих пір, поки величина наступного члена не стане меншою за визначену точність (epsilon = 0.000001).
2. **Сума перших 10 членів ряду:** Програма також обчислює суму перших 10 членів ряду для порівняння результатів.

Ряд визначений формулою: , де n — індекс члена ряду.

***Основні функції програми:***

**1.Ітеративний підрахунок суми з точністю:** У циклі do-while програма додає члени ряду до загальної суми, поки абсолютне значення наступного члена ряду не стане менше epsilon (0.000001). Це дозволяє обчислити суму нескінченного ряду з заданою точністю.

**2. Підрахунок суми перших 10 членів:** Використовуючи цикл for, програма обчислює і виводить суму перших 10 членів ряду.

***Ієрархія та структура класів:***

**Клас task\_07:** Основний клас програми, який містить метод Main для виконання обчислень.

* **Метод Main:**
  + Обчислює суму ряду двома способами:
    1. Обчислює суму з точністю до 0.000001, використовуючи цикл do-while.
    2. Обчислює суму перших 10 членів ряду за допомогою циклу for.
  + Виводить результати на екран.

**Завдання 8\***

Виконати минуле завдання, переробивши алгоритм вираховування суми ряду таким чином:

* похибка та кількість членів ряду мають вводитися як вхідні параметри у відповідні змінні;
* сума ряду вираховується або при досягненні заданої похибки, або при досягненні заданої кількості членів ряду із видачою відповідного повідомлення.

***Текст програми:***

using System;

class task\_08

{

static void Main()

{

// Введення похибки та кількості членів ряду

Console.Write("Введіть похибку: ");

double epsilon = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть максимальну кількість членів ряду: ");

int maxTerms = Convert.ToInt32(Console.ReadLine());

double sum = 0.0;

double term;

int n = 0;

// Обчислення суми членів ряду з заданою похибкою або кількістю членів

do

{

term = Math.Pow(-1, n) \* ((n + 1) / (Math.Pow(2, n - 1)));

sum += term;

n++;

} while (Math.Abs(term) > epsilon && n < maxTerms);

// Видача відповідного повідомлення

if (Math.Abs(term) <= epsilon)

{

Console.WriteLine("Сума членів ряду з точністю до " + epsilon + ": " + sum);

}

else

{

Console.WriteLine("Досягнуто максимальну кількість членів ряду (" + maxTerms + "): " + sum);

}

// Обчислення суми перших 10 членів ряду

sum = 0.0;

for (n = 0; n < 10; n++)

{

term = Math.Pow(-1, n) \* ((n + 1) / (Math.Pow(2, n - 1)));

sum += term;

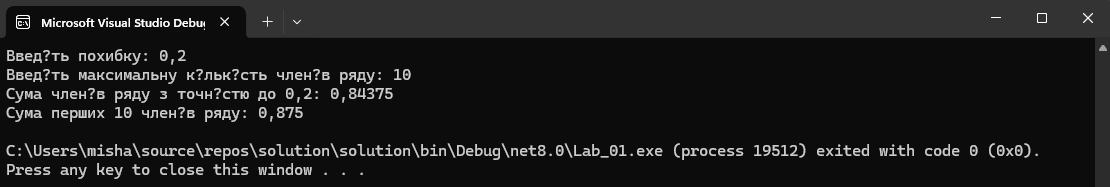
}

Console.WriteLine("Сума перших 10 членів ряду: " + sum);

}

}

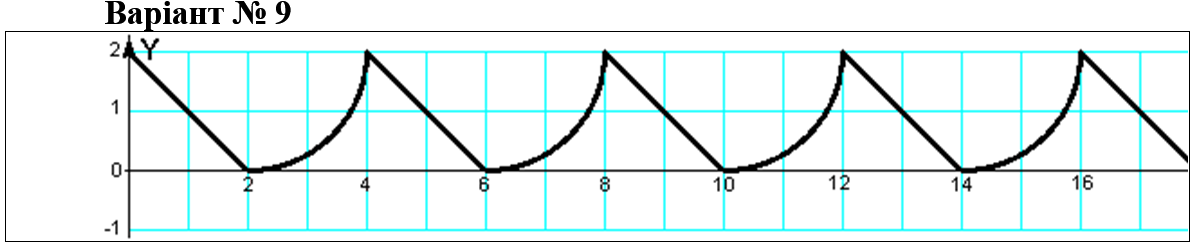
***Результат роботи програми:***



**Завдання 9**

Використовуючи завдання власного варіанта до лабораторної роботи 7 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 7:* *Для функції* ***y = f****(****x****), графік якої наведений у виданому індивідуальному завданні, вивести на екран значення* ***y*** *для* ***x =*** *0, 0.25, 0.5 ... 19.75. Необов’язкова для виконання додаткова вимога – отримати на екрані графік функції з використанням засобів текстового режиму.*

******

***Текст програми:***

using System;

class task\_09

{

static void Main()

{

double minX = 0;

double maxX = 19.75;

double step = 0.25;

int width = 80; // Ширина графіка

int height = 20; // Висота графіка

char[,] graph = new char[height, width];

// Ініціалізація графіка пробілами

for (int i = 0; i < height; i++)

{

for (int j = 0; j < width; j++)

{

graph[i, j] = ' ';

}

}

// Обчислення значень функції та заповнення графіка

for (double x = minX; x <= maxX; x += step)

{

double y = SawtoothWave(x);

Console.WriteLine($"x = {x:F2}, y = {y:F2}");

int graphX = (int)((x - minX) / (maxX - minX) \* (width - 1));

int graphY = (int)((y + 1) / 2 \* (height - 1));

graph[height - 1 - graphY, graphX] = '\*';

}

// Виведення графіка на екран

for (int i = 0; i < height; i++)

{

for (int j = 0; j < width; j++)

{

Console.Write(graph[i, j]);

}

Console.WriteLine();

}

}

static double SawtoothWave(double x)

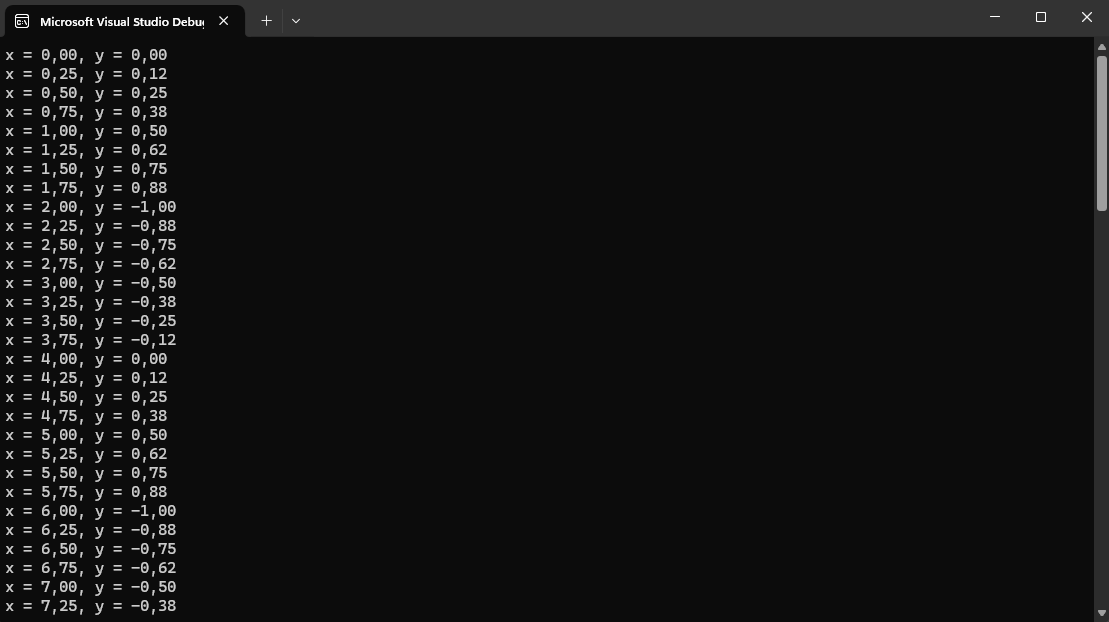
{

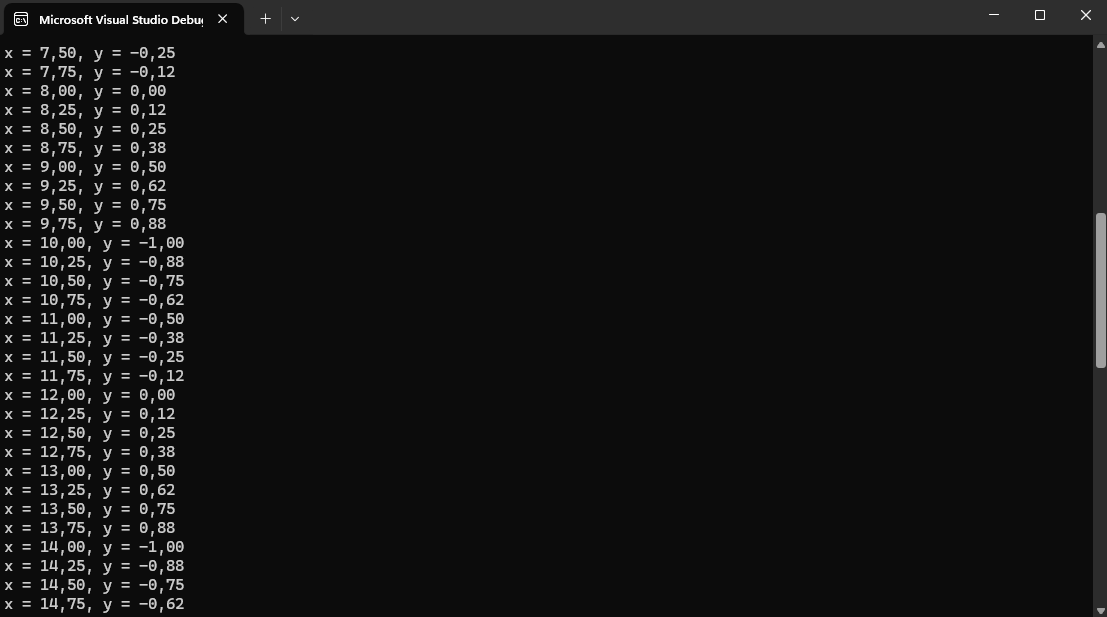
return 2 \* (x / 4.0 - Math.Floor(x / 4.0 + 0.5));

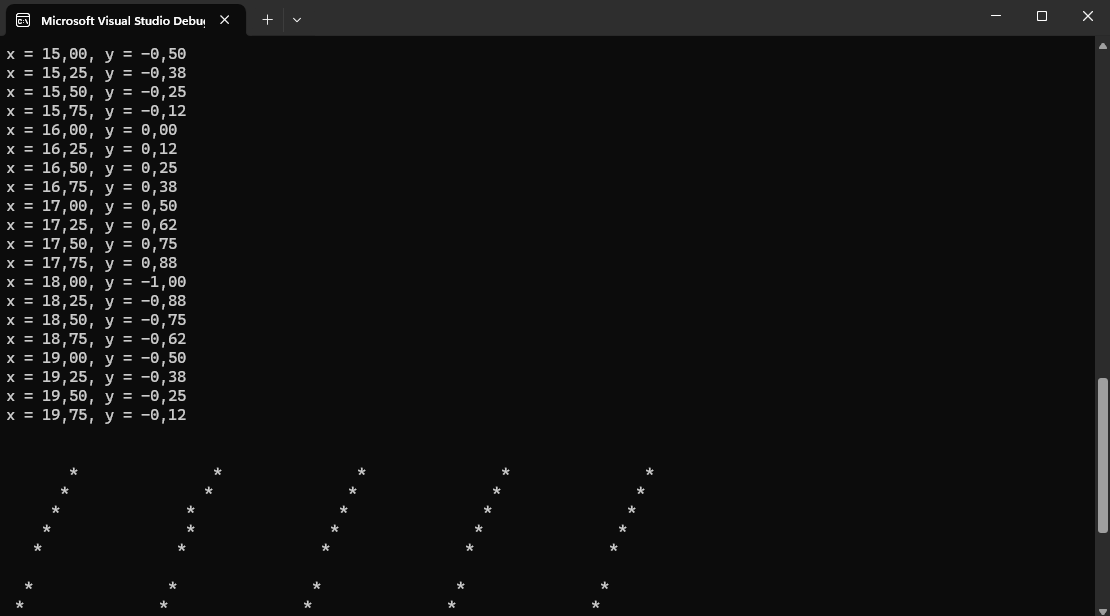
}

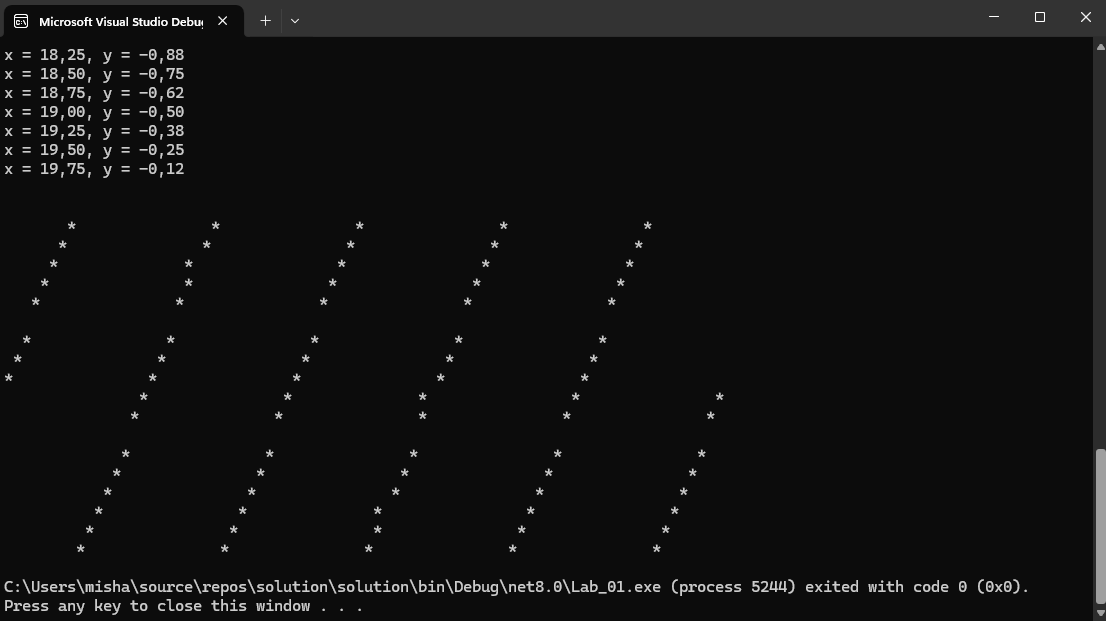
}

***Результат роботи програми:***





****

****

***Опис програми:***

Ця програма на C# генерує та виводить графік пилообразної (зубчатої) хвилі у текстовій формі в консолі. Вона обчислює значення пилообразної функції в межах від 0 до 19.75 з кроком 0.25 та відображає це на графіку розміром 80 на 20 символів.

***Основні функції програми:***

**1.** **Ініціалізація графіка:** Створюється двовимірний масив символів graph, що буде використовуватись для побудови графіка. Масив заповнюється пробілами.

**2.** **Обчислення значень функції:** Для кожного значення xxx з кроком 0.25 обчислюється відповідне значення пилообразної хвилі yyy, використовуючи функцію SawtoothWave. Результати перетворюються на координати графіка і позначаються символом \* в масиві graph.

**3.** **Виведення графіка:** Після того, як масив заповнений символами, графік виводиться на екран рядок за рядком.

***Ієрархія та структура класів:***

**Клас task\_09:** Основний клас програми, містить метод Main для управління логікою програми та статичний метод SawtoothWave, що реалізує обчислення пилообразної функції.

* **Метод Main:**
  + Ініціалізує графік.
  + Обчислює значення функції пилообразної хвилі для кожного значення xxx.
  + Позначає точки на графіку та виводить графік на екран.
* **Метод SawtoothWave(double x):**
  + Обчислює значення пилообразної функції для заданого значення xxx.

**Завдання 10\***

Виконати минуле завдання, покращивши зовнішній вигляд побудованого графіку за допомогою символів псевдографіки ⸜, ⸝, ⸺, \, /, ­ та ін.) та налаштування кольорів консолі, наприклад, числа відобразити одним кольором, пояснюючий текст – іншим, графік – іншим від попередніх.

***Текст програми:***

using System;

class task\_10

{

static void Main()

{

double minX = 0;

double maxX = 19.75;

double step = 0.25;

int width = 80; // Ширина графіка

int height = 20; // Висота графіка

char[,] graph = new char[height, width];

// Ініціалізація графіка пробілами

for (int i = 0; i < height; i++)

{

for (int j = 0; j < width; j++)

{

graph[i, j] = ' ';

}

}

// Обчислення значень функції та заповнення графіка

for (double x = minX; x <= maxX; x += step)

{

double y = SawtoothWave(x);

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine($"x = {x:F2}, y = {y:F2}");

Console.ResetColor();

int graphX = (int)((x - minX) / (maxX - minX) \* (width - 1));

int graphY = (int)((y + 1) / 2 \* (height - 1));

graph[height - 1 - graphY, graphX] = '⸺';

}

// Виведення графіка на екран

Console.ForegroundColor = ConsoleColor.Cyan;

for (int i = 0; i < height; i++)

{

for (int j = 0; j < width; j++)

{

Console.Write(graph[i, j]);

}

Console.WriteLine();

}

Console.ResetColor();

}

static double SawtoothWave(double x)

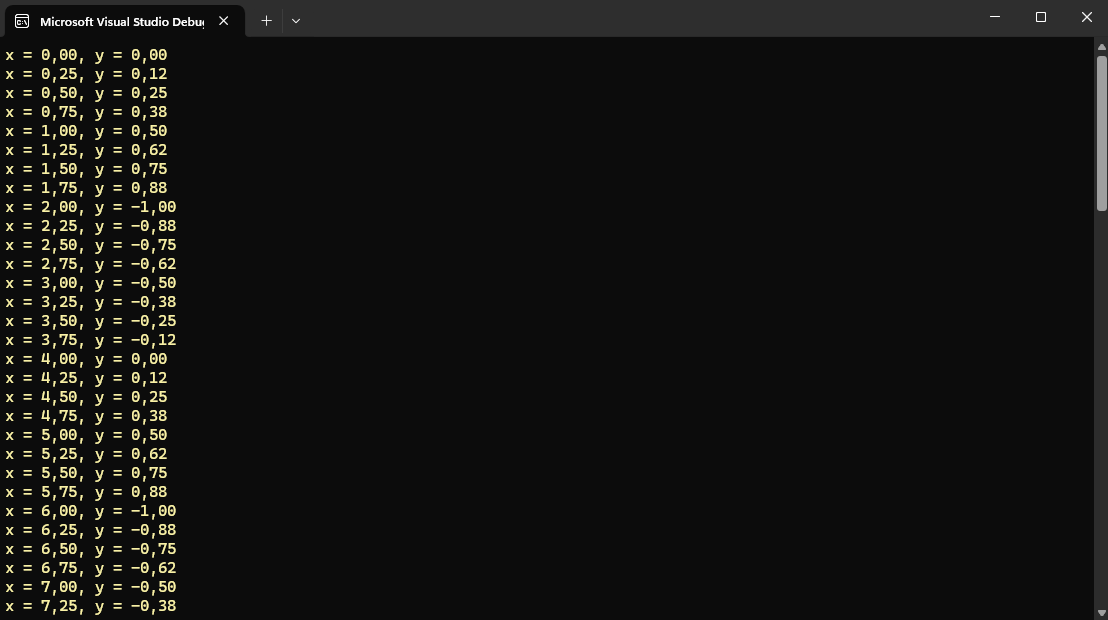
{

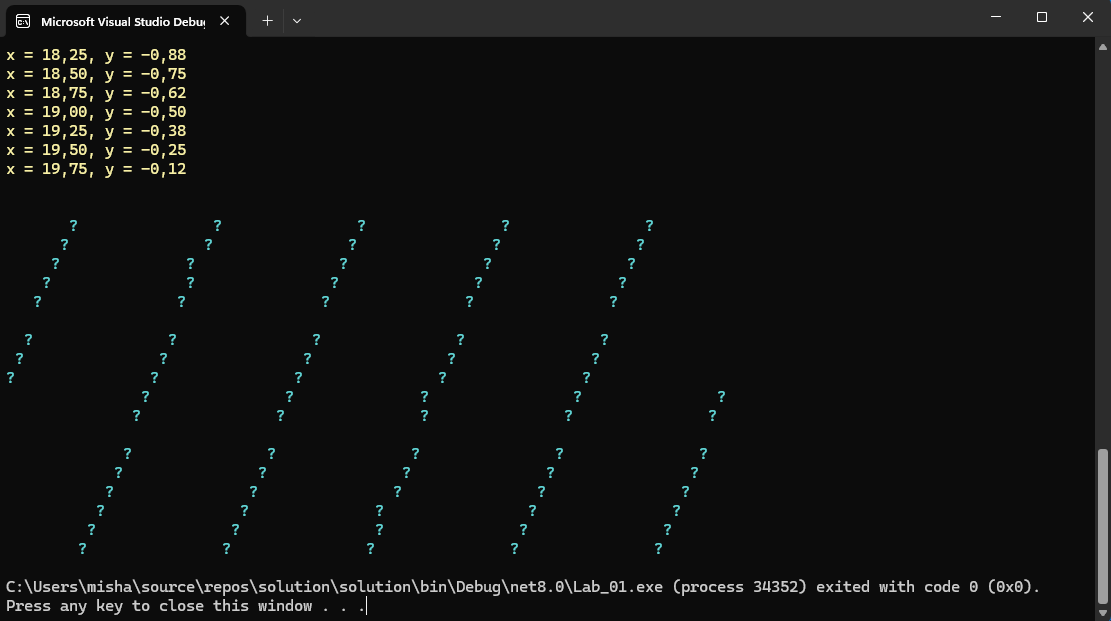
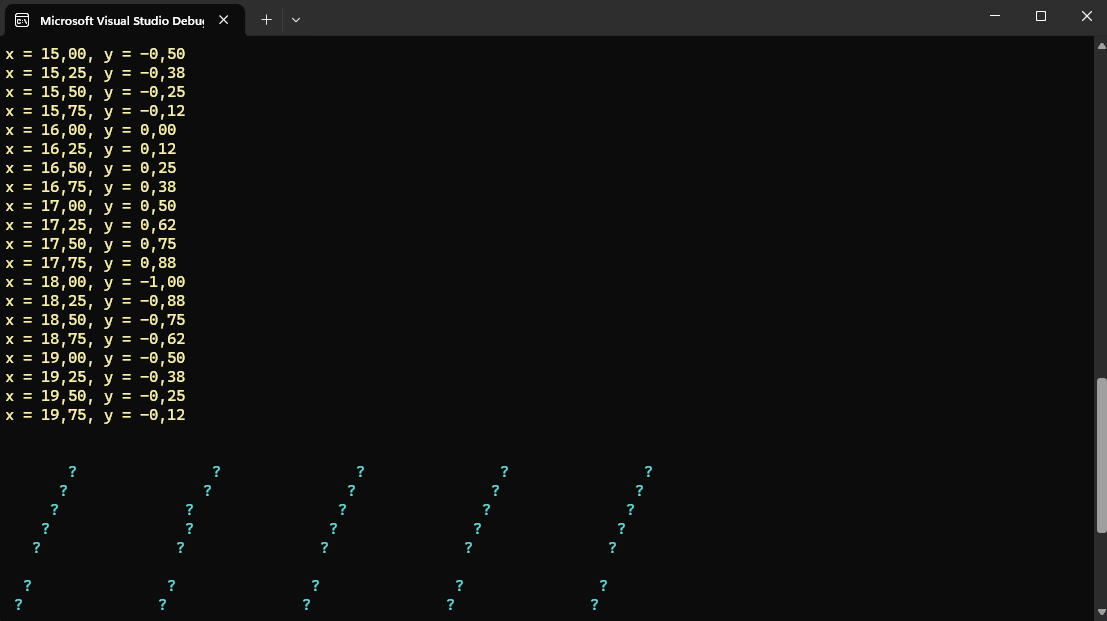
return 2 \* (x / 4.0 - Math.Floor(x / 4.0 + 0.5));

}

}

***Результат роботи програми:***





**Висновок**

У ході виконання лабораторної роботи було отримано практичні навички в написанні програм на мові C# з використанням базових типів даних, операторів, а також засобів введення та виведення інформації в консолі. Було успішно розроблено програми, які демонструють застосування різних структур управління (цикли, умовні оператори), математичних операцій та функцій для обчислення значень, і виведення результатів у текстовій формі.

Ця робота дозволила закріпити розуміння основ програмування в C#, що є важливим кроком у подальшому навчанні програмуванню та розробці програмного забезпечення.