Exercise 1: The Geometry Calculator

Goal: Practice returning named tuples from a method.

Write a method called CalculateRectangle that takes two double parameters: length and width. The method should return a **named tuple** containing both the **Area** and the **Perimeter** of the rectangle.

- **Formula:** $Area = length \times width$
- **Formula:** $Perimeter = 2 \times (length + width)$

Exercise 2: The "Min-Max" Array Finder

Goal: Practice iterating through data to populate a tuple.

Create a method called GetLimits that accepts an int[] array. It should return a tuple containing the **minimum** and the **maximum** values found in that array.

- *Bonus:* Handle the case where the array might be empty by returning (0, 0) or using nullable types.

Exercise 3: Quick Swap Logic

Goal: Use tuple syntax to swap variables without a temporary "temp" variable.

Given two integer variables, int a = 10; and int b = 20;, use a single line of tuple assignment to swap their values so that a becomes 20 and b becomes 10. Print the results to verify.

Exercise 4: Parsing Sensitive Data (Deconstruction)

Goal: Practice deconstruction and the "discard" symbol.

Imagine a method exists with this signature:

static (string FullName, string Email, string SSN) GetUserData()

Call this method and **deconstruct** the result into local variables. However, you only care about the FullName and Email. Use the **discard operator (_)** to ignore the SSN (Social Security Number) so it isn't stored in a variable.

Exercise 5: List of Tuples (Inventory Management)

Goal: Manage a collection of tuples.

Create a List that stores tuples representing store inventory. Each tuple should hold:

1. string ProductName
2. int Quantity
3. decimal Price

Add three items to the list. Then, use a foreach loop to iterate through the list and print a formatted string for each item:

*"Product: [Name] | Total Value: [Quantity * Price]"*