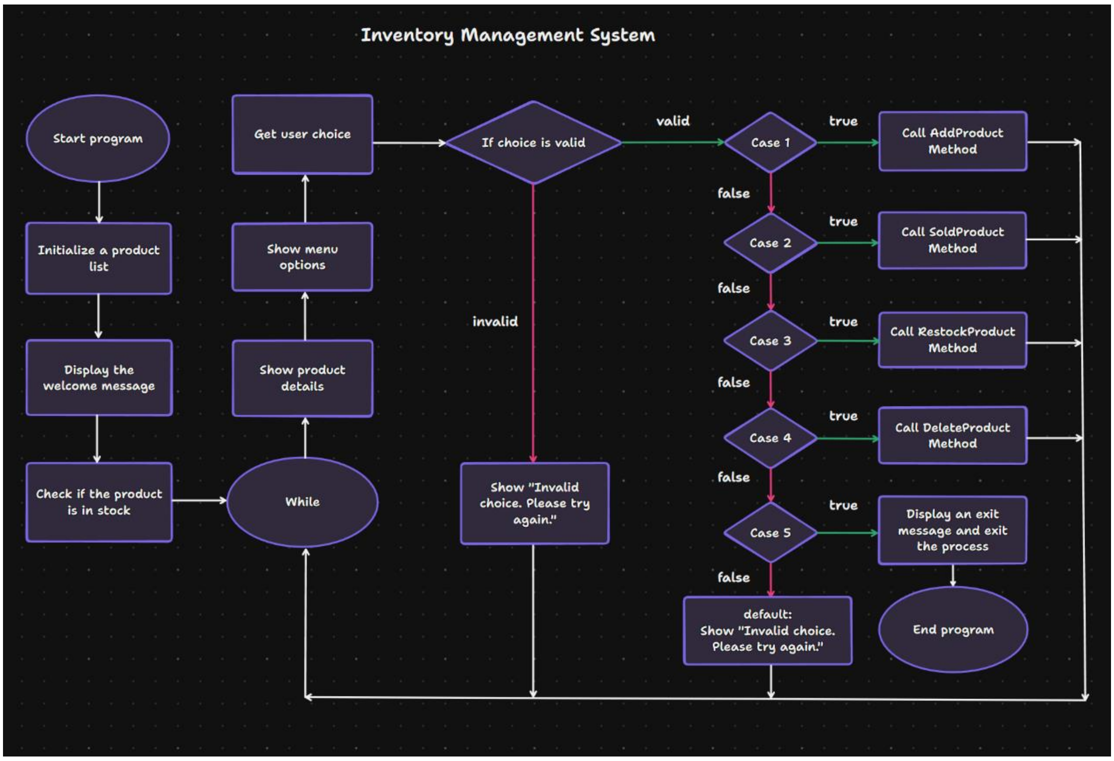
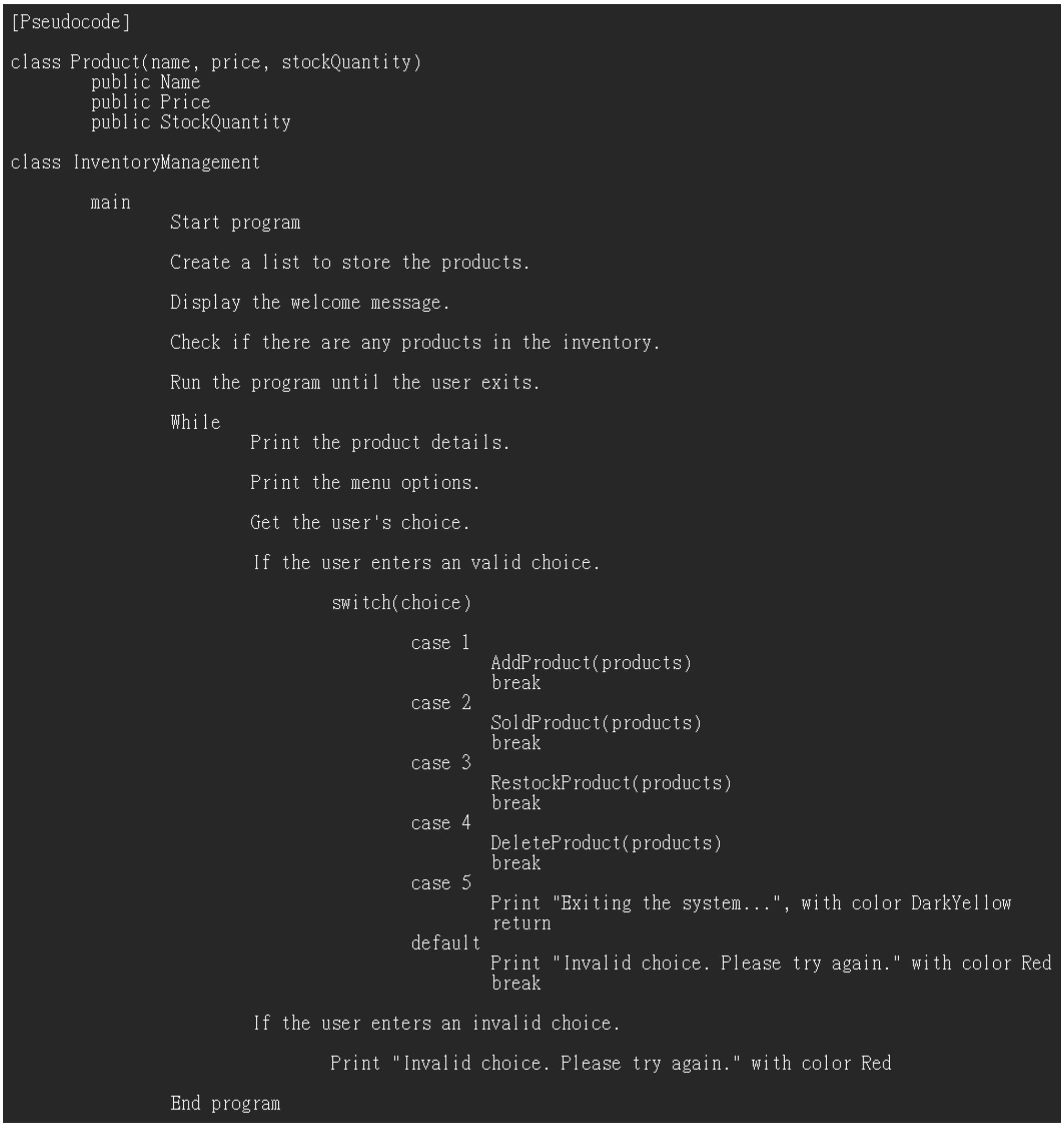
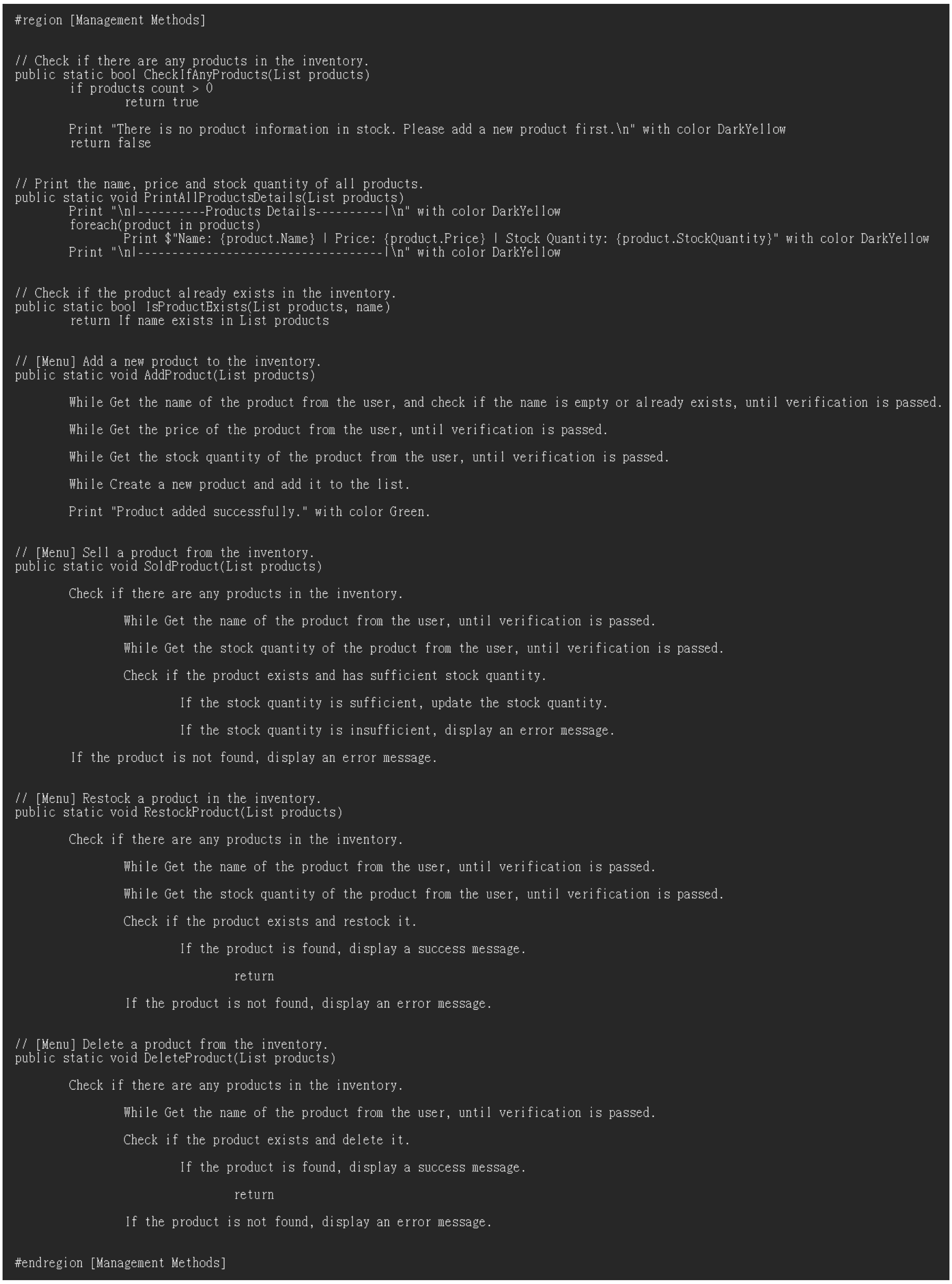
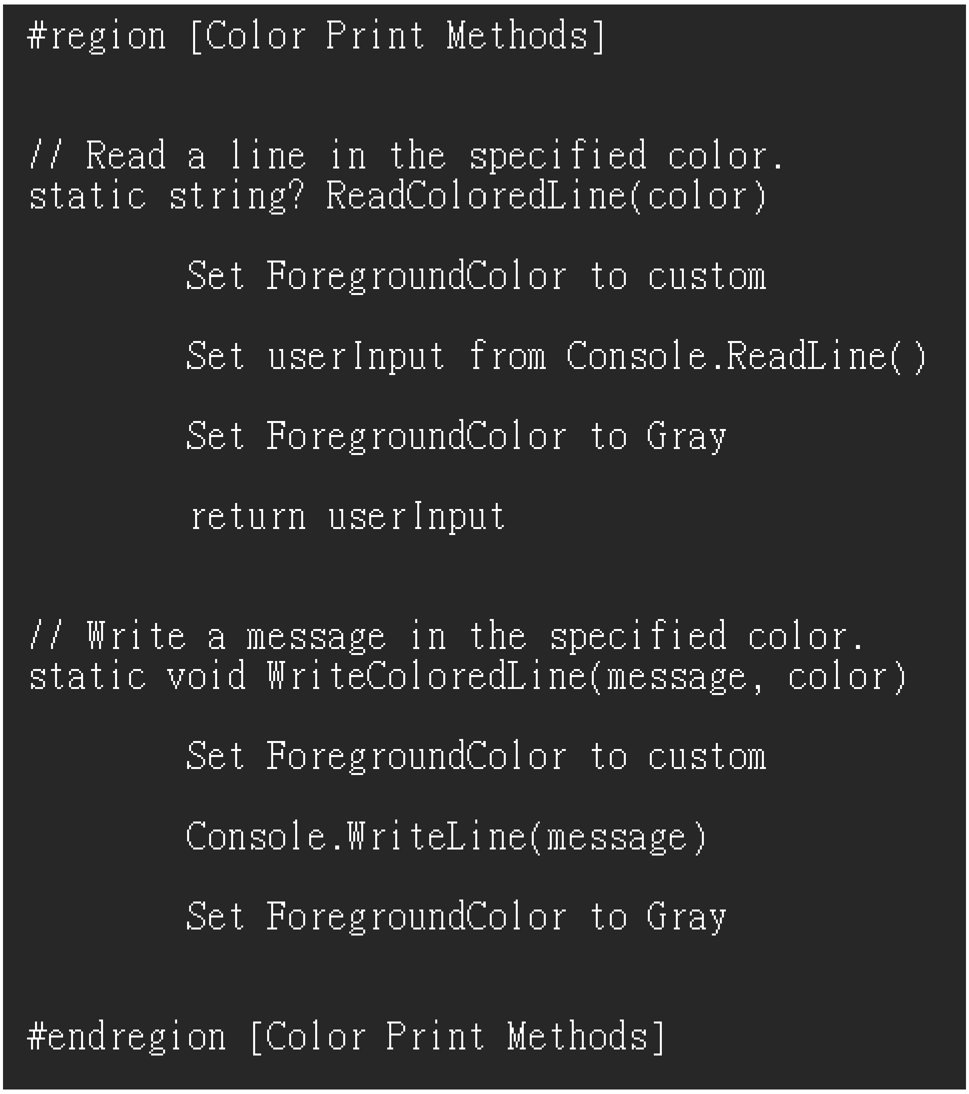

Inventory Management System
Start program
Initialize a product list
Display the welcome message
Check if the product is in stock
While
Show product details
Show menu options
Get user choice
If choice is valid
invalid
Show "Invalid choice. Please try again."
valid
Case 1
true
Call AddProduct Method
false
Case 2
true
Call SoldProduct Method
false
Case 3
true
Call RestockProduct Method
false
Case 4
true
Call DeleteProduct Method
false
Case 5
true
Display an exit message and exit the process
End program
false
default:
Show "Invalid choice. Please try again."



```
[Pseudocode]

class Product(name, price, stockQuantity)
        public Name
        public Price
        public StockQuantity

class InventoryManagement

        main
                Start program

                Create a list to store the products.

                Display the welcome message.

                Check if there are any products in the inventory.

                Run the program until the user exits.

                While
                        Print the product details.

                        Print the menu options.

                        Get the user's choice.

                        If the user enters an valid choice.

                                switch(choice)

                                        case 1
                                                AddProduct(products)
                                                break
                                        case 2
                                                SoldProduct(products)
                                                break
                                        case 3
                                                RestockProduct(products)
                                                break
                                        case 4
                                                DeleteProduct(products)
                                                break
                                        case 5
                                                Print "Exiting the system...", with color DarkYellow
                                                return
                                        default
                                                Print "Invalid choice. Please try again." with color Red
                                                break

                        If the user enters an invalid choice.

                                Print "Invalid choice. Please try again." with color Red

                End program
```



```
#region [Management Methods]


// Check if there are any products in the inventory.
public static bool CheckIfAnyProducts(List products)
        if products count > 0
                return true

        Print "There is no product information in stock. Please add a new product first.\n" with color DarkYellow
        return false


// Print the name, price and stock quantity of all products.
public static void PrintAllProductsDetails(List products)
        Print "\n|----------Products Details----------|\n" with color DarkYellow
        foreach(product in products)
                Print $"Name: {product.Name} | Price: {product.Price} | Stock Quantity: {product.StockQuantity}" with color DarkYellow
        Print "\n|------------------------------------|\n" with color DarkYellow


// Check if the product already exists in the inventory.
public static bool IsProductExists(List products, name)
        return If name exists in List products


// [Menu] Add a new product to the inventory.
public static void AddProduct(List products)

        While Get the name of the product from the user, and check if the name is empty or already exists, until verification is passed.

        While Get the price of the product from the user, until verification is passed.

        While Get the stock quantity of the product from the user, until verification is passed.

        While Create a new product and add it to the list.

        Print "Product added successfully." with color Green.


// [Menu] Sell a product from the inventory.
public static void SoldProduct(List products)

        Check if there are any products in the inventory.

                While Get the name of the product from the user, until verification is passed.

                While Get the stock quantity of the product from the user, until verification is passed.

                Check if the product exists and has sufficient stock quantity.

                        If the stock quantity is sufficient, update the stock quantity.

                        If the stock quantity is insufficient, display an error message.

        If the product is not found, display an error message.


// [Menu] Restock a product in the inventory.
public static void RestockProduct(List products)

        Check if there are any products in the inventory.

                While Get the name of the product from the user, until verification is passed.

                While Get the stock quantity of the product from the user, until verification is passed.

                Check if the product exists and restock it.

                        If the product is found, display a success message.

                                return

                If the product is not found, display an error message.


// [Menu] Delete a product from the inventory.
public static void DeleteProduct(List products)

        Check if there are any products in the inventory.

                While Get the name of the product from the user, until verification is passed.

                Check if the product exists and delete it.

                        If the product is found, display a success message.

                                return

                If the product is not found, display an error message.


#endregion [Management Methods]
```



```
#region [Color Print Methods]


// Read a line in the specified color.
static string? ReadColoredLine(color)

        Set ForegroundColor to custom

        Set userInput from Console.ReadLine()

        Set ForegroundColor to Gray

        return userInput


// Write a message in the specified color.
static void WriteColoredLine(message, color)

        Set ForegroundColor to custom

        Console.WriteLine(message)

        Set ForegroundColor to Gray


#endregion [Color Print Methods]
```