**Exercise 1: Inventory Management System**

**Scenario:**

You are developing an inventory management system for a warehouse. Efficient data storage and retrieval are crucial.

**Steps:**

1. **Understand the Problem:**
   * Explain why data structures and algorithms are essential in handling large inventories.
   * Discuss the types of data structures suitable for this problem.
2. **Setup:**
   * Create a new project for the inventory management system.
3. **Implementation:**
   * Define a class Product with attributes like **productId**, **productName**, **quantity**, and **price**.
   * Choose an appropriate data structure to store the products (e.g., ArrayList, HashMap).
   * Implement methods to add, update, and delete products from the inventory.
4. **Analysis:**
   * Analyze the time complexity of each operation (add, update, delete) in your chosen data structure.
   * Discuss how you can optimize these operations.

**CODE:**

using System;

class Product

{

public int ProductId { get; set; }

public string ProductName { get; set; }

public int Quantity { get; set; }

public decimal Price { get; set; }

public Product(int productId, string productName, int quantity, decimal price)

{

ProductId = productId;

ProductName = productName;

Quantity = quantity;

Price = price;

}

}

class Inventory

{

private Dictionary<int, Product> products = new Dictionary<int, Product>();

public void AddProduct(Product product) => products[product.ProductId] = product;

public void UpdateProduct(Product product)

{

if (products.ContainsKey(product.ProductId))

products[product.ProductId] = product;

}

public void DeleteProduct(int productId) => products.Remove(productId);

public void DisplayInventory()

{

foreach (var product in products.Values)

Console.WriteLine($"{product.ProductId} - {product.ProductName}, Qty: {product.Quantity}, Price: {product.Price}");

}

}

class Program1

{

static void Main()

{

var inventory = new Inventory();

inventory.AddProduct(new Product(1, "Laptop", 10, 50000));

inventory.AddProduct(new Product(2, "Mouse", 50, 500));

inventory.UpdateProduct(new Product(2, "Mouse", 60, 550));

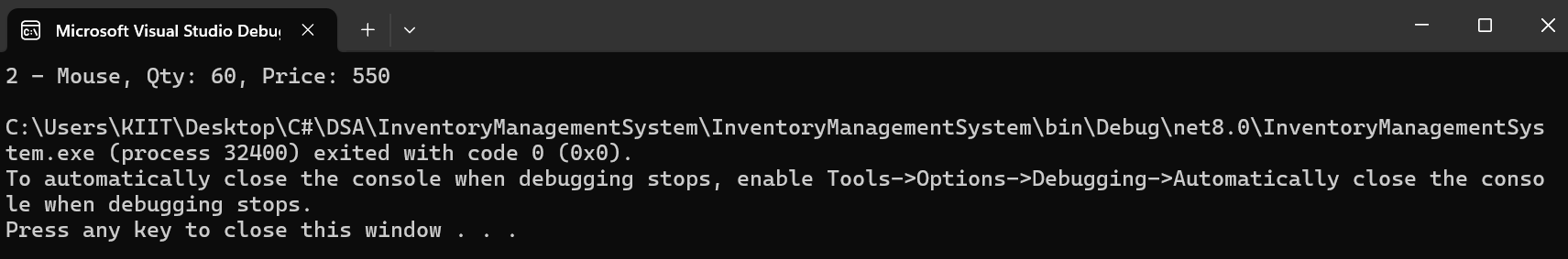
inventory.DeleteProduct(1);

inventory.DisplayInventory();

}

}

**OUTPUT:**



**ANALYSIS:**

****Time Complexity:****

* **Add**: O(1)
* **Update**: O(1)
* **Delete**: O(1)