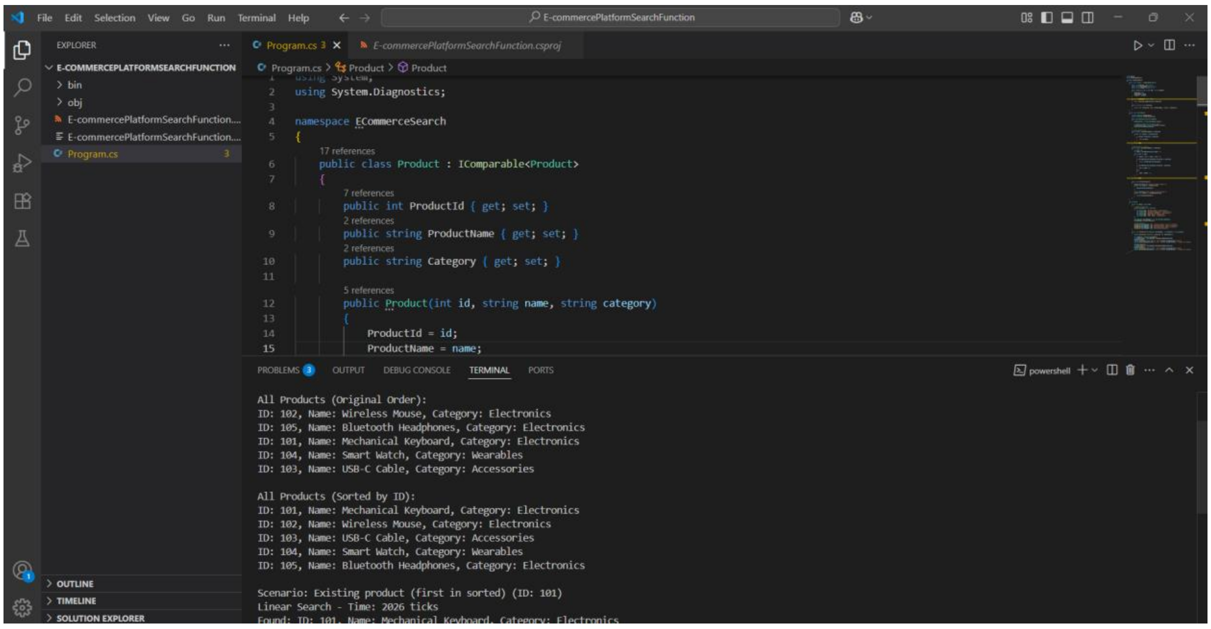
E-commerce Platform Search Function

```
using System;

using System.Diagnostics;


namespace ECommerceSearch

{

   public class Product : IComparable<Product>

   {

      public int ProductId { get; set; }

      public string ProductName { get; set; }

      public string Category { get; set; }


      public Product(int id, string name, string category)

      {

         ProductId = id;

         ProductName = name;

         Category = category;

      }


      public int CompareTo(Product other)

      {

         return ProductId.CompareTo(other.ProductId);

      }


      public override string ToString()

      {

         return $"ID: {ProductId}, Name: {ProductName}, Category: {Category}";

      }

   }


   public class SearchManager
```

```
{
    private Product[] productsArray;
    private Product[] sortedProductsArray;

    public SearchManager(Product[] products)
    {
        productsArray = (Product[])products.Clone();

        sortedProductsArray = (Product[])products.Clone();
        Array.Sort(sortedProductsArray);
    }

    public Product LinearSearchById(int productId)
    {
        foreach (var product in productsArray)
        {
            if (product.ProductId == productId)
            {
                return product;
            }
        }
        return null;
    }


    public Product BinarySearchById(int productId)
    {
        int left = 0;
        int right = sortedProductsArray.Length - 1;

        while (left <= right)
    {
        int middle = left + (right - left) / 2;

        if (sortedProductsArray[middle].ProductId == productId)
        {
            return sortedProductsArray[middle];
        }

        if (sortedProductsArray[middle].ProductId < productId)
        {
            left = middle + 1;
        }
        else
        {
            right = middle - 1;
        }
    }

    return null;
}

public void PrintAllProducts()
{
    Console.WriteLine("All Products (Original Order):");
    foreach (var product in productsArray)
    {
        Console.WriteLine(product);
    }

    Console.WriteLine("\nAll Products (Sorted by ID):");
    foreach (var product in sortedProductsArray)
```

```
        {
            Console.WriteLine(product);
        }
    }
}

class Program
{
    static void Main(string[] args)
    {

        Product[] products = new Product[]
        {
            new Product(102, "Wireless Mouse", "Electronics"),
            new Product(105, "Bluetooth Headphones", "Electronics"),
            new Product(101, "Mechanical Keyboard", "Electronics"),
            new Product(104, "Smart Watch", "Wearables"),
            new Product(103, "USB-C Cable", "Accessories")
        };

        SearchManager searchManager = new SearchManager(products);
        searchManager.PrintAllProducts();

        TestSearch(searchManager, 101, "Existing product (first in sorted)");
        TestSearch(searchManager, 103, "Existing product (middle in sorted)");
        TestSearch(searchManager, 105, "Existing product (last in sorted)");
        TestSearch(searchManager, 999, "Non-existing product");
    }

    static void TestSearch(SearchManager searchManager, int productId, string scenario)
    {
            Console.WriteLine($"\nScenario: {scenario} (ID: {productId})");


            var stopwatch = Stopwatch.StartNew();
            var productLinear = searchManager.LinearSearchById(productId);
            stopwatch.Stop();
            Console.WriteLine($"Linear Search - Time: {stopwatch.ElapsedTicks} ticks");
            Console.WriteLine(productLinear != null ? $"Found: {productLinear}" : "Product not found");


            stopwatch.Restart();
            var productBinary = searchManager.BinarySearchById(productId);
            stopwatch.Stop();
            Console.WriteLine($"Binary Search - Time: {stopwatch.ElapsedTicks} ticks");
            Console.WriteLine(productBinary != null ? $"Found: {productBinary}" : "Product not found");
        }
    }
}
```


File Edit Selection View Go Run Terminal Help
E-commercePlatformSearchFunction
EXPLORER
E-COMMERCEPLATFORMSEARCHFUNCTION
bin
obj
E-commercePlatformSearchFunction....
E-commercePlatformSearchFunction....
Program.cs 3
Program.cs 3
E-commercePlatformSearchFunction.csproj
Program.cs > Product > Product
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
OUTLINE
TIMELINE
SOLUTION EXPLORER

```
using System;
using System.Diagnostics;

namespace ECommerceSearch
{
    17 references
    public class Product : IComparable<Product>
    {
        7 references
        public int ProductId { get; set; }
        2 references
        public string ProductName { get; set; }
        2 references
        public string Category { get; set; }

        5 references
        public Product(int id, string name, string category)
        {
            ProductId = id;
            ProductName = name;
```

```
All Products (Original Order):
ID: 102, Name: Wireless Mouse, Category: Electronics
ID: 105, Name: Bluetooth Headphones, Category: Electronics
ID: 101, Name: Mechanical Keyboard, Category: Electronics
ID: 104, Name: Smart Watch, Category: Wearables
ID: 103, Name: USB-C Cable, Category: Accessories

All Products (Sorted by ID):
ID: 101, Name: Mechanical Keyboard, Category: Electronics
ID: 102, Name: Wireless Mouse, Category: Electronics
ID: 103, Name: USB-C Cable, Category: Accessories
ID: 104, Name: Smart Watch, Category: Wearables
ID: 105, Name: Bluetooth Headphones, Category: Electronics

Scenario: Existing product (first in sorted) (ID: 101)
Linear Search - Time: 2026 ticks
Found: ID: 101, Name: Mechanical Keyboard, Category: Electronics
```