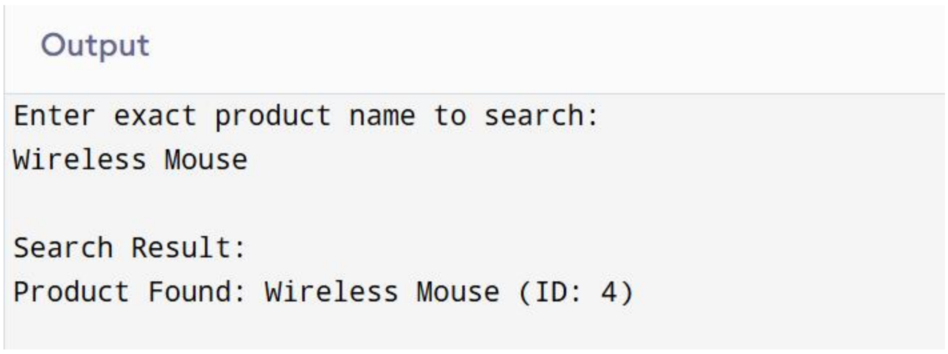
DSA

Exercise 2: E-commerce Platform Search Function

```
using System;
using System.Collections.Generic;

namespace ECommerceExactSearch
{

   public class Product
   {
      public int Id { get; }
      public string Name { get; }

      public Product(int id, string name)
      {
         Id = id;
         Name = name;
      }
   }

   public class ECommercePlatform
   {
      private Dictionary<string, Product> productDictionary;

      public ECommercePlatform()
      {
         productDictionary = new Dictionary<string, Product>(StringComparer.OrdinalIgnoreCase)
         {
            { "Laptop", new Product(1, "Laptop") },
            { "Smartphone", new Product(2, "Smartphone") },
            { "Bluetooth Speaker", new Product(3, "Bluetooth Speaker") },
            { "Wireless Mouse", new Product(4, "Wireless Mouse") },
            { "Laptop Stand", new Product(5, "Laptop Stand") },
            { "USB-C Cable", new Product(6, "USB-C Cable") }
         };
      }

      public Product SearchExact(string productName)
      {
         if (productDictionary.TryGetValue(productName, out Product product))
         {
            return product;
         }
         return null;
      }
   }

   class Program
   {
      static void Main(string[] args)
      {
         ECommercePlatform platform = new ECommercePlatform();
```

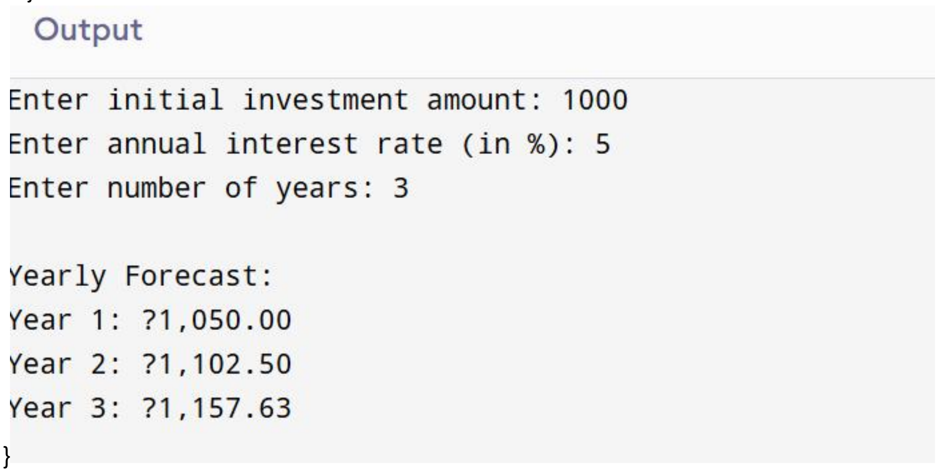
```
            Console.WriteLine("Enter exact product name to search:");
            string query = Console.ReadLine();

            var result = platform.SearchExact(query);

            Console.WriteLine("\nSearch Result:");
            if (result != null)
            {
                Console.WriteLine($"Product Found: {result.Name} (ID: {result.Id})");
            }
            else
            {
                Console.WriteLine("Product not found.");
            }
        }
    }
}
```



```
Enter exact product name to search:
Wireless Mouse

Search Result:
Product Found: Wireless Mouse (ID: 4)
```

<u>Exercise 7: Financial Forecasting</u>

```
using System;
using System.Collections.Generic;

class FinancialForecast
{
    static void Main()
    {

        Console.Write("Enter initial investment amount: ");
        double principal = Convert.ToDouble(Console.ReadLine());

        Console.Write("Enter annual interest rate (in %): ");
        double annualRate = Convert.ToDouble(Console.ReadLine()) / 100;

        Console.Write("Enter number of years: ");
        int years = Convert.ToInt32(Console.ReadLine());

        List<double> yearlyBalances = new List<double>();

        for (int year = 1; year <= years; year++)
        {
            principal = principal * (1 + annualRate);
            yearlyBalances.Add(principal);
        }

        Console.WriteLine("\nYearly Forecast:");
        for (int i = 0; i < yearlyBalances.Count; i++)
        {
            Console.WriteLine($"Year {i + 1}: {yearlyBalances[i]:C2}");
        }
    }
```



Output

```
Enter initial investment amount: 1000
Enter annual interest rate (in %): 5
Enter number of years: 3

Yearly Forecast:
Year 1: ?1,050.00
Year 2: ?1,102.50
Year 3: ?1,157.63
```

```
}
```