**Week1 Algorithms\_Data Structures**

| Exercise 2: E-commerce Platform Search Function  Code:  using System;  public class product  {  private int productid;  private string productname;  private string category;  public product(int productid, string productname, string category)  {  this.productid = productid;  this.productname = productname;  this.category = category;  }  public int getproductid() { return productid; }  public string getproductname() { return productname; }  public string getcategory() { return category; }  public override string ToString()  {  return productid + ": " + productname + " (" + category + ")";  }  }  public class searchengine  {  public static product linearsearch(product[] products, string name)  {  foreach (product p in products)  {  if (string.Equals(p.getproductname(), name, StringComparison.OrdinalIgnoreCase))  {  return p;  }  }  return null;  }  public static product binarysearch(product[] products, string name)  {  int left = 0;  int right = products.Length - 1;  while (left <= right)  {  int mid = (left + right) / 2;  int cmp = string.Compare(products[mid].getproductname(), name, true);  if (cmp == 0) return products[mid];  else if (cmp < 0) left = mid + 1;  else right = mid - 1;  }  return null;  }  public static void sortproductsbyname(product[] products)  {  Array.Sort(products, (p1, p2) => string.Compare(p1.getproductname(), p2.getproductname(), true));  }  }  class program  {  static void Main(string[] args)  {  product[] products = {  new product(3, "Phone", "Electronics"),  new product(1, "Laptop", "Electronics"),  new product(2, "Tablet", "Electronics"),  };  // Test linear search  product result1 = searchengine.linearsearch(products, "Tablet");  Console.WriteLine(result1 != null ? "Found via linear search: " + result1 : "Not found");  // Test binary search  searchengine.sortproductsbyname(products);  product result2 = searchengine.binarysearch(products, "Laptop");  Console.WriteLine(result2 != null ? "Found via binary search: " + result2 : "Not found");  }  }  Output: |
| --- |

| Exercise 7: Financial Forecasting |
| --- |

Code:

using System;

class FinancialForecast

{

public static double FutureValue(double presentValue, double rate, int years)

{

if (years <= 0)

return presentValue;

return FutureValue(presentValue \* (1 + rate), rate, years - 1);

}

static void Main(string[] args)

{

double presentValue = 1000.0; // Starting amount

double rate = 0.05; // 5% annual growth

int years = 10;

double result = FutureValue(presentValue, rate, years);

Console.WriteLine($"Future value after {years} years at {rate\*100}% rate is: {result:F2}");

}

}

Output:

