Practical Questions Part I

1. Write a C# program to implement the Bubble Sort algorithm.
2. Write a C# program to implement the Selection Sort algorithm.
3. Write a C# program to implement the Merge Sort algorithm.
4. Write a C# program to perform a Linear Search on an array of integers.
5. Write a C# program to perform a Binary Search on a sorted array of integers.

Practical Questions Part II

Sample Class with Data

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

public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Category { get; set; }
    public decimal Price { get; set; }
}

public class Program
{
    public static void Main()
    {
        List<Product> products = new List<Product>
        {
            new Product { Id = 1, Name = "Laptop", Category = "Electronics", Price = 1000 },
            new Product { Id = 2, Name = "Smartphone", Category = "Electronics", Price = 700 },
            new Product { Id = 3, Name = "Table", Category = "Furniture", Price = 150 },
            new Product { Id = 4, Name = "Chair", Category = "Furniture", Price = 85 },
            new Product { Id = 5, Name = "Pen", Category = "Stationery", Price = 2 },
            new Product { Id = 6, Name = "Notebook", Category = "Stationery", Price = 5 }
        };

        // Add LINQ queries here to answer the questions
    }
}

```

1. Select products that belong to the "Electronics" category.
2. Select only the names of all products.
3. Assuming a nested list structure, select all characters in the names of products.
4. Group products by their category.
5. Order products by their price.
6. Order products by category, then by price.
7. Order products by price in descending order.
8. Order products by category, then by price in descending order.
9. Convert the product names to a list.
10. Convert the product names to an array.
11. Combine two lists of products and remove duplicates.
12. Find products that are in the original list but not in another list.
13. Select distinct categories from the product list.
14. Find common products between two lists.
15. Get the first product in the list or default if the list is empty.
16. Get the last product in the list or default if the list is empty.
17. Get a single product with a specific name or default if not found.
18. Check if the product list contains a specific product.
19. Concatenate all product names into a single string separated by commas.
20. Count the number of products in the "Furniture" category.
21. Calculate the total price of all products.
22. Calculate the average price of all products.
23. Find the product with the minimum price.
24. Find the product with the maximum price.
25. Check if there are any products in the "Stationery" category.