Άσκηση 1: Ταξινόμηση Απλού Πίνακα

Περιγραφή: Ταξινομήστε έναν απλό πίνακα ακεραίων χρησιμοποιώντας την ταξινόμηση συγχώνευσης.

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 38, 27, 43, 3, 9, 82, 10 };
        MergeSort(array, 0, array.Length - 1);

        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void MergeSort(int[] array, int left, int right)
    {
        if (left < right)
        {
            int middle = (left + right) / 2;

            MergeSort(array, left, middle);
            MergeSort(array, middle + 1, right);

            Merge(array, left, middle, right);
        }
    }

    static void Merge(int[] array, int left, int middle, int right)
    {
        int n1 = middle - left + 1;
        int n2 = right - middle;

        int[] leftArray = new int[n1];
        int[] rightArray = new int[n2];

        Array.Copy(array, left, leftArray, 0, n1);
        Array.Copy(array, middle + 1, rightArray, 0, n2);

        int i = 0, j = 0, k = left;
        while (i < n1 && j < n2)
        {
            if (leftArray[i] <= rightArray[j])
            {
```

```
            array[k] = leftArray[i];
            i++;
        }
        else
        {
            array[k] = rightArray[j];
            j++;
        }
        k++;
    }

    while (i < n1)
    {
        array[k] = leftArray[i];
        i++;
        k++;
    }

    while (j < n2)
    {
        array[k] = rightArray[j];
        j++;
        k++;
    }
  }
}
```

Άσκηση 2: Ταξινόμηση Πίνακα με Αρνητικούς Αριθμούς

Περιγραφή: Ταξινομήστε έναν πίνακα που περιέχει και θετικούς και αρνητικούς αριθμούς.

```
using System;

class Program
{
  static void Main()
  {
    int[] array = { 12, -3, 5, 0, -9, 8, -2 };
    MergeSort(array, 0, array.Length - 1);

    Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
  }

  static void MergeSort(int[] array, int left, int right)
  {
    if (left < right)
```

```
    {
        int middle = (left + right) / 2;

        MergeSort(array, left, middle);
        MergeSort(array, middle + 1, right);

        Merge(array, left, middle, right);
    }
}

static void Merge(int[] array, int left, int middle, int right)
{
    int n1 = middle - left + 1;
    int n2 = right - middle;

    int[] leftArray = new int[n1];
    int[] rightArray = new int[n2];

    Array.Copy(array, left, leftArray, 0, n1);
    Array.Copy(array, middle + 1, rightArray, 0, n2);

    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2)
    {
        if (leftArray[i] <= rightArray[j])
        {
            array[k] = leftArray[i];
            i++;
        }
        else
        {
            array[k] = rightArray[j];
            j++;
        }
        k++;
    }

    while (i < n1)
    {
        array[k] = leftArray[i];
        i++;
        k++;
    }

    while (j < n2)
    {
```

```
            array[k] = rightArray[j];
            j++;
            k++;
        }
    }
}
```

Άσκηση 3: Ταξινόμηση Σειράς Αλφαριθμητικών

Περιγραφή: Ταξινομήστε έναν πίνακα αλφαριθμητικών (strings) με την ταξινόμηση συγχώνευσης

```
using System;

class Program
{
    static void Main()
    {
        string[] array = { "apple", "orange", "banana", "pear", "grape" };
        MergeSort(array, 0, array.Length - 1);

        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void MergeSort(string[] array, int left, int right)
    {
        if (left < right)
        {
            int middle = (left + right) / 2;

            MergeSort(array, left, middle);
            MergeSort(array, middle + 1, right);

            Merge(array, left, middle, right);
        }
    }

    static void Merge(string[] array, int left, int middle, int right)
    {
        int n1 = middle - left + 1;
        int n2 = right - middle;

        string[] leftArray = new string[n1];
        string[] rightArray = new string[n2];

        Array.Copy(array, left, leftArray, 0, n1);
        Array.Copy(array, middle + 1, rightArray, 0, n2);
```

```
        int i = 0, j = 0, k = left;
        while (i < n1 && j < n2)
        {
            if (string.Compare(leftArray[i], rightArray[j]) <= 0)
            {
                array[k] = leftArray[i];
                i++;
            }
            else
            {
                array[k] = rightArray[j];
                j++;
            }
            k++;
        }

        while (i < n1)
        {
            array[k] = leftArray[i];
            i++;
            k++;
        }

        while (j < n2)
        {
            array[k] = rightArray[j];
            j++;
            k++;
        }
    }
}
```

Άσκηση 4: Ταξινόμηση Πίνακα με Διπλούς Αριθμούς

Περιγραφή: Ταξινομήστε έναν πίνακα που περιέχει διπλούς αριθμούς (double).

```
using System;

class Program
{
    static void Main()
    {
        double[] array = { 2.5, -1.1, 3.3, 0.0, 4.4, -2.2, 1.1 };
        MergeSort(array, 0, array.Length - 1);

        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }
```

```
static void MergeSort(double[] array, int left, int right)
{
    if (left < right)
    {
        int middle = (left + right) / 2;

        MergeSort(array, left, middle);
        MergeSort(array, middle + 1, right);

        Merge(array, left, middle, right);
    }
}

static void Merge(double[] array, int left, int middle, int right)
{
    int n1 = middle - left + 1;
    int n2 = right - middle;

    double[] leftArray = new double[n1];
    double[] rightArray = new double[n2];

    Array.Copy(array, left, leftArray, 0, n1);
    Array.Copy(array, middle + 1, rightArray, 0, n2);

    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2)
    {
        if (leftArray[i] <= rightArray[j])
        {
            array[k] = leftArray[i];
            i++;
        }
        else
        {
            array[k] = rightArray[j];
            j++;
        }
        k++;
    }

    while (i < n1)
    {
        array[k] = leftArray[i];
        i++;
        k++;
```

```
        }

        while (j < n2)
        {
            array[k] = rightArray[j];
            j++;
            k++;
        }
    }
}
```

Άσκηση 5: Ταξινόμηση Πίνακα με Επαναλαμβανόμενες Τιμές

Περιγραφή: Ταξινομήστε έναν πίνακα που περιέχει επαναλαμβανόμενες τιμές.

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 5, 3, 8, 3, 2, 7, 1, 5, 6 };
        MergeSort(array, 0, array.Length - 1);

        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void MergeSort(int[] array, int left, int right)
    {
        if (left < right)
        {
            int middle = (left + right) / 2;

            MergeSort(array, left, middle);
            MergeSort(array, middle + 1, right);

            Merge(array, left, middle, right);
        }
    }

    static void Merge(int[] array, int left, int middle, int right)
    {
        int n1 = middle - left + 1;
        int n2 = right - middle;

        int[] leftArray = new int[n1];
```

```
        int[] rightArray = new int[n2];

        Array.Copy(array, left, leftArray, 0, n1);
        Array.Copy(array, middle + 1, rightArray, 0, n2);

        int i = 0, j = 0, k = left;
        while (i < n1 && j < n2)
        {
            if (leftArray[i] <= rightArray[j])
            {
                array[k] = leftArray[i];
                i++;
            }
            else
            {
                array[k] = rightArray[j];
                j++;
            }
            k++;
        }

        while (i < n1)
        {
            array[k] = leftArray[i];
            i++;
            k++;
        }

        while (j < n2)
        {
            array[k] = rightArray[j];
            j++;
            k++;
        }
    }
}
```

Άσκηση 6: Ταξινόμηση Μεγάλου Πίνακα

Περιγραφή: Ταξινομήστε έναν μεγάλο πίνακα ακεραίων (1000+ στοιχεία) με την ταξινόμηση συγχώνευσης.

```
using System;

class Program
{
    static void Main()
    {
        Random rand = new Random();
        int[] array = new int[1000];
        for (int i = 0; i < array.Length; i++)
        {
            array[i] = rand.Next(1, 1001);
        }

        MergeSort(array, 0, array.Length - 1);

        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void MergeSort(int[] array, int left, int right)
    {
        if (left < right)
        {
            int middle = (left + right) / 2;

            MergeSort(array, left, middle);
            MergeSort(array, middle + 1, right);

            Merge(array, left, middle, right);
        }
    }

    static void Merge(int[] array, int left, int middle, int right)
    {
        int n1 = middle - left + 1;
        int n2 = right - middle;

        int[] leftArray = new int[n1];
        int[] rightArray = new int[n2];

        Array.Copy(array, left, leftArray, 0, n1);
        Array.Copy(array, middle + 1, rightArray, 0, n2);

        int i = 0, j = 0, k = left;
        while (i < n1 && j < n2)
        {
            if (leftArray[i] <= rightArray[j])
```

```
            {
                array[k] = leftArray[i];
                i++;
            }
            else
            {
                array[k] = rightArray[j];
                j++;
            }
            k++;
        }

        while (i < n1)
        {
            array[k] = leftArray[i];
            i++;
            k++;
        }

        while (j < n2)
        {
            array[k] = rightArray[j];
            j++;
            k++;
        }
    }
}
```