Answer on Question #44589, Programming, C#

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

namespace Merge_Sort
{
    class Program
    {
        /// <summary>
        /// Main function
        /// </summary>
        /// <param name="args"></param>
        static void Main(string[] args)
        {
            //promt user to enter the size of array
            Console.Write("Enter size of array: ");
            //read size from keyboard
            int size = int.Parse(Console.ReadLine());
            //array for values
            int[] numbers = new int[size];
            //promt user to enter values
            for (int i = 0; i < size; i++)
            {
                Console.Write("Enter value " + (i + 1).ToString()+": ");
                //read values from keyboard
                numbers[i] = int.Parse(Console.ReadLine());
            }
            //show all values
            Console.WriteLine("Your array before sorting:");
            for (int k = 0; k < size; k++)
            {
                //print values
                Console.Write(numbers[k] + " ");
            }

            Console.WriteLine("\n\nYour array after sorting");
            //call function Merge
            Merge(numbers, 0, size - 1);
            for (int i = 0; i < size; i++)
            {
                Console.Write(numbers[i] + " ");
            }
            //delay
            Console.ReadLine();

        }
        /// <summary>
        /// Merge Sort function
        /// </summary>
        /// <param name="input"></param>
        /// <param name="l"></param>
        /// <param name="r"></param>
        public static void Merge(int[] arrayofnumbers , int l, int r)
        {
            if (l < r)
            {
                //calculate middle index of array
```

```
                int mid = (l + r) / 2;
                //call function Merge
                Merge(arrayofnumbers, l, mid);
                Merge(arrayofnumbers, mid + 1, r);

                //Merge
                int[] leftArray = new int[mid - l + 1];
                int[] rightArray = new int[r - mid];
                //copyt array
                Array.Copy(arrayofnumbers, l, leftArray, 0, mid - l + 1);
                Array.Copy(arrayofnumbers, mid + 1, rightArray, 0, r - mid);

                int row = 0;
                int column = 0;

                //loop array
                for (int k = l; k < r + 1; k++)
                {
                    if (row == leftArray.Length)
                    {
                        //add value to right
                        arrayofnumbers[k] = rightArray[column];
                        column++;
                    }
                    else if (column == rightArray.Length)
                    {
                        //add value to left
                        arrayofnumbers[k] = leftArray[row];
                        row++;
                    }
                    else if (leftArray[row] <= rightArray[column])
                    {
                        //add value to left
                        arrayofnumbers[k] = leftArray[row];
                        row++;
                    }
                    else
                    {
                        //add value to right
                        arrayofnumbers[k] = rightArray[column];
                        column++;
                    }
                }
            }
        }
    }
}
```