REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 5 |
| Tema a Evaluar | Mezcla Ascendente |
| Fecha de Entrega | 09 de Noviembre |

**Programa Fuente**

| using System;  namespace P40\_Mezcla\_Ascendente  {  internal class Program  {  //Inicio declaración método ProcedimientoMerge  public static void ProcedimientoMerge(ref int[] ArregloOriginal, int Front, int Rear)  {  //Inicio bloque de instrucciones  if (Front != Rear)  {  int half = (Front + Rear) / 2;  ProcedimientoMerge(ref ArregloOriginal, Front, half);  ProcedimientoMerge(ref ArregloOriginal, half + 1, Rear);  int[] auxy = FuncionMerge(ArregloOriginal, Front, half, half + 1, Rear);  Array.Copy(auxy,0,ArregloOriginal,Front, auxy.Length);  }  //Fin bloque de instrucciones  }  //Fin declaración método ProcedimientoMerge  //Inicio declaración método FuncionMerge  private static int[] FuncionMerge(int[] ArregloOriginal, int FrontOne, int RearOne, int FrontTwo, int RearTwo)  {  //Inicio declaración de variables  int First = FrontOne, Second = FrontTwo;  int[] MergeArray = new int[RearOne - FrontOne + RearTwo - FrontTwo + 2];  //Fin declaración de variables  //Inicio bloque de instrucciones  for (int i = 0; i < MergeArray.Length; i++)  {  if(Second != MergeArray.Length)  {  if(First> RearOne && Second <= RearTwo)  {  MergeArray[i] = ArregloOriginal[Second];  Second++;  }  if(Second > RearTwo && First <= RearOne)  {  MergeArray[i] = ArregloOriginal[First];  First++;  }  if(First <= RearOne && Second <= RearTwo)  {  if (ArregloOriginal[Second] <= ArregloOriginal[First])  {  MergeArray[i] = ArregloOriginal[Second];  Second++;  }  else  {  MergeArray[i] = ArregloOriginal[First];  First++;  }  }  }  else  {  if(First <= RearOne)  {  MergeArray[i] = ArregloOriginal[First];  First++;  }  }  }  return MergeArray;  //Fin bloque de instrucciones  }  //Fin declaración método FuncionMerge  //Inicio declaración método MostrarArreglo  public static void MostrarArreglo(int[] Arreglo)  {  for (int i = 0; i < Arreglo.Length; i++)  {  if (i == Arreglo.Length / 2) Console.Write("\n\t");  Console.Write($"[{Arreglo[i]}]");  }  }  //Fin declaración método MostrarArreglo  static void Main(string[] args)  {  Console.Title = "P40 Mezcla Ascendente";  //Inicio declaración de variables  int[] ArregloNumerosControl = new int[30];  //Fin declaración de variables  Console.WriteLine("\n\t\t.: CAPTURANDO NÚMEROS DE CONTROL :.");  for (int i = 0; i < ArregloNumerosControl.Length; i++)  {  Console.Write($"\n\n\t[{i + 1}] - Ingrese un número de control: ");  ArregloNumerosControl[i] = int.Parse(Console.ReadLine());  Console.Write($"\n\tEl número de control \"{ArregloNumerosControl[i]}\" ha sido capturada correctamente");  }  Console.Write("\n\n\tLos números de control han sido capturados correctamente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  Console.Write("\n\n\t\t.: MOSTRANDO ARREGLO ORIGINAL :.\n\n\t");  //Llamada a método para mostrar arreglo  MostrarArreglo(ArregloNumerosControl);  //Llamada a método para ordenar  ProcedimientoMerge(ref ArregloNumerosControl, 0, ArregloNumerosControl.Length - 1);  Console.Write("\n\n\t\t.: MOSTRANDO ARREGLO ORDENADO ASCENDENTE :.\n\n\t");  MostrarArreglo(ArregloNumerosControl);  Console.Write("\n\n\tLos números de control han sido ordenados correctamente" +  "\n\n\tPresione la tecla <ESC> para salir del menú...");  while (Console.ReadKey().Key != ConsoleKey.Escape) ; Console.Clear();  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

**EXPLICA BREVEMENTE QUÉ TE PARECIÓ EL PROBLEMA, QUE SE TE DIFICULTO Y COMO LO SOLUCIONASTE.**

| Al igual que los demás, no hubo complicación pues solamente se copió la estructura vista en clase y explicada por la profesora.  La estructura ya era ordenamiento ascendente así que no hubo modificaciones.  Cometí el error de capturar algunos números de control de 8 dígitos y otros de 9 dígitos, así que eso dificulta un poco más darse cuenta de que están ordenados, pero el código funciona perfectamente. |
| --- |