REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 5 |
| Tema a Evaluar | Intercalación Simple Ascendente Método 2 |
| Fecha de Entrega | 08 de Noviembre del 2023 |

**Programa Fuente**

| using System;  namespace P39\_Intercalacion\_simple\_Ascendente\_Metodo\_2  {  internal class Program  {  //Inicio declaración método IntercalacionSimpleM2  public static void IntercalacionSimpleM2(int[] NumberOneArray, int[] NumberTwoArray, ref int[] ResultArray)  {  //Inicio declaración de variables locales  int TotalElementsArrayOne = NumberOneArray.Length;  int TotalElementsArrayTwo = NumberTwoArray.Length;  int i = 0, j = 0, k = 0;  //Fin declaración de variables locales  //Inicio bloque de instrucciones  while (i < TotalElementsArrayOne && j < TotalElementsArrayTwo)  {  if (NumberOneArray[i] <= NumberTwoArray[j])  {  ResultArray[k] = NumberOneArray[i];  i++;  }  else  {  ResultArray[k] = NumberTwoArray[j];  j++;  }  k++;  }  if(i == TotalElementsArrayOne)  {  for(int p = j; p < TotalElementsArrayTwo; p++)  {  ResultArray[k] = NumberTwoArray[p];  k++;  }  }  else  {  for(int p = i; p < TotalElementsArrayOne; p++)  {  ResultArray[k] = NumberOneArray[p];  k++;  }  }  //Fin bloque de instrucciones  }  //Fin declaración método IntercalacionSimpleM2  //Declaración de método quicksort  public static int[] QuickSort(int[] Numeros, int primero, int ultimo)  {  int i, j, central, pivote, temp;  central = (primero + ultimo) / 2;  pivote = Numeros[central];  i = primero; j = ultimo;  do  {  while (Numeros[i] < pivote) i++;  while (Numeros[j] > pivote) j--;  if (i <= j)  {  temp = Numeros[i];  Numeros[i] = Numeros[j];  Numeros[j] = temp;  i++; j--;  }  } while (i <= j);  if (primero < j)  {  Numeros = QuickSort(Numeros, primero, j);  }  if (i < ultimo)  {  Numeros = QuickSort(Numeros, i, ultimo);  }  return Numeros;  }  //Fin declaración de método quicksort  //Inicio declaración método MostrarArreglo  public static void MostrarArreglo(int[] Arreglo)  {  for (int i = 0; i < Arreglo.Length; i++)  {  Console.Write($"[{Arreglo[i]}]");  }  }  //Fin declaración método MostrarArreglo  //Inicio declaración método MostrarArreglo sobrecarga para mostrar en doble renglón el arreglo  public static void MostrarArreglo(int[] Arreglo, bool DobleLine)  {  if(DobleLine)  for (int i = 0; i < Arreglo.Length; i++)  {  if (i == (Arreglo.Length / 2) + 1) Console.Write("\n\t");  Console.Write($"[{Arreglo[i]}]");  }  }  //Fin declaración método MostrarArreglo sobrecarga para mostrar en doble renglón el arreglo  static void Main(string[] args)  {  Console.Title = "P39 Intercalación Simple Ascendente Método 2";  //Inicio declaración de variables  int[] RandomArrayNumberOne = new int[15];  int[] RandomArrayNumberTwo = new int[11];  int[] IntercalationResultArray = new int[RandomArrayNumberOne.Length + RandomArrayNumberTwo.Length];  Random RandomNumber = new Random();  int TemporalRandom;  //Fin declaración de variables  //Inicio captura datos arreglo 1  Console.WriteLine("\n\t\t.: CAPTURANDO ARREGLO 1 :." +  "\n\n\t| Se insertaran números aleatorios a este arreglo |");  for (int i = 0; i < RandomArrayNumberOne.Length; i++)  {  TemporalRandom = RandomNumber.Next(0, 9999);  Console.Write($"\n\n\t[{i + 1}] - Ingrese un número: {TemporalRandom}");  RandomArrayNumberOne[i] = TemporalRandom;  Console.Write($"\n\tEl número {RandomArrayNumberOne[i]} ha sido capturado correctamente");  }  Console.Write("\n\n\tLos números han sido capturados correctamente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Fin captura datos arreglo 1  //Inicio captura datos arreglo 2  Console.WriteLine("\n\t\t.: CAPTURANDO ARREGLO 2 :." +  "\n\n\t| Se insertaran números aleatorios a este arreglo |");  for (int i = 0; i < RandomArrayNumberTwo.Length; i++)  {  TemporalRandom = RandomNumber.Next(0, 9999);  Console.Write($"\n\n\t[{i + 1}] - Ingrese un número: {TemporalRandom}");  RandomArrayNumberTwo[i] = TemporalRandom;  Console.Write($"\n\tEl número {RandomArrayNumberTwo[i]} ha sido capturado correctamente");  }  Console.Write("\n\n\tLos números han sido capturados correctamente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Fin captura datos arreglo 2  Console.Write("\n\n\t\t.: MOSTRANDO ARREGLOS ORIGINALES :.");  //Llamada a métodos para mostrar arreglo  Console.Write("\n\n\t\t| Arreglo 1 |\n\n\t");  MostrarArreglo(RandomArrayNumberOne);  Console.Write("\n\n\t\t| Arreglo 2 |\n\n\t");  MostrarArreglo(RandomArrayNumberTwo);  Console.Write("\n\n\n\t\t.: MOSTRANDO ARREGLOS ASCENDENTEMENTE :.");  //Llamada a métodos para ordenar Quicksort  RandomArrayNumberOne = QuickSort(RandomArrayNumberOne, 0, RandomArrayNumberOne.Length - 1);  RandomArrayNumberTwo = QuickSort(RandomArrayNumberTwo, 0, RandomArrayNumberTwo.Length - 1);  Console.Write("\n\n\t\t| Arreglo 1 |\n\n\t");  MostrarArreglo(RandomArrayNumberOne);  Console.Write("\n\n\t\t| Arreglo 2 |\n\n\t");  MostrarArreglo(RandomArrayNumberTwo);    Console.Write("\n\n\t\tLos números han sido mostrados correctamente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Llamada a método para ordenar  IntercalacionSimpleM2(RandomArrayNumberOne, RandomArrayNumberTwo, ref IntercalationResultArray);  Console.Write("\n\n\t\t.: MOSTRANDO RESULTADOS :.");  Console.Write("\n\n\n\t\t| Arreglo 1 |\n\n\t");  MostrarArreglo(RandomArrayNumberOne);  Console.Write("\n\n\t\t| Arreglo 2 |\n\n\t");  MostrarArreglo(RandomArrayNumberTwo);  Console.Write("\n\n\t\t| Arreglo Resultante |\n\n\t");  MostrarArreglo(IntercalationResultArray, true);  Console.Write("\n\n\tLos números han sido ordenados correctamente" +  "\n\n\tPresione la tecla <ESC> para salir del programa...");  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.**

| El problema me resultó más sencillo de entender que el anterior, debido a que es como saltarse todos los pasos hasta el último. Es una gran ventaja que los dos arreglos empleados ya estén previamente ordenados.  Utilicé QuickSort para ordenar los primeros arreglos individuales y después los ordené los dos en un tercer arreglo con ayuda del segundo método de Intercalación simple.  También utilicé sobrecarga de métodos para poder mostrar el tercer arreglo en dos líneas distintas debido a que este es el doble de longitud aproximadamente y quedaba demasiado grande. |
| --- |