REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 5 |
| Tema a Evaluar | Burbuja Optimizada Ascendente |
| Fecha de Entrega | 01 de Noviembre del 2023 |

**Programa Fuente**

| using System;  namespace P33\_Burbuja\_Optimizada\_Ascendente  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Declaración de método para ordenar  public static string[] OrdenamientoBurbujaOptimizada(string[] Apodos)  { //Declaración de variables locales  bool flag = true;  string t;  //Fin variables locales  for (int i = 0; i < Apodos.Length - 1 && flag == true; i++)  {  flag = false;  for (int j = 0; j < Apodos.Length - i - 1; j++)  {  if (Apodos[j].CompareTo(Apodos[j + 1]) > 0)  {  flag = true;  t = Apodos[j];  Apodos[j] = Apodos[j + 1];  Apodos[j + 1] = t;  }  }  }  return Apodos;  }  //Fin declaración de método para ordenar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio declaración método para mostrar  public static void Mostrar(string[] Apodos)  {  for (int i = 1; i <= Apodos.Length; i++)  {  Console.Write($"[{Apodos[i - 1]}] ");  }  }  //Fin declaración de método mostrar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  static void Main(string[] args)  {  Console.Title = "P33 Burbuja Optimizada Ascendente";  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Declaración de variables locales  string[] Apodos = new string[20];  string Apodo; char opcion;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  do //Do while para controlar menú  {  //Despliegue de menú  Console.Write("\n\t\t.: MENÚ DE OPCIONES :." +  "\n\n\t[ 1 ] - Capturar Apodos" +  "\n\t[ 2 ] - Ordenar Apodos Ascendientemente" +  "\n\t[ 3 ] - Salir del Programa" +  "\n\n\tIngrese el número de la opción deseada: ");  if (char.TryParse(Console.ReadLine(), out opcion) != false) //Try catch para controlar errores de formato  {  Console.Write($"\n\n\tLa opción {opcion} se ha capturado correctamente" +  $"\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Switch case para controlar opciones  switch (opcion)  {  case '1':  {  Console.WriteLine("\n\t.: CAPTURA DE DATOS :.");  for (int i = 1; i <= Apodos.Length; i++) //Ciclo para capturar todos los apodos  {  Console.Write($"\n\t{i}.- Ingrese un apodo: "); Apodo = Console.ReadLine();  Console.WriteLine($"\n\tSe ha capturado el apodo \"{Apodo}\" correctamente.");  Apodos[i - 1] = Apodo;  }  Console.Write("\n\tCaptura de datos terminada, presione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '2':  {  Console.WriteLine("\n\t\t.: ORDENANDO DATOS ASCENDIENTEMENTE :." +  "\n\n\t| Este es el arreglo original sin ordenar |\n\t");  Mostrar(Apodos); //Llamada a método para mostrar  Console.WriteLine("\n\n\t| Este es el arreglo ordenado ascendientemente |\n\t");  Apodos = OrdenamientoBurbujaOptimizada(Apodos);//llamada a método para ordenar  Mostrar(Apodos);  Console.Write("\n\n\tOrdenamiento terminado, presione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '3':  {  Console.Write("\n\t\t:. SALIDA :." +  "\n\n\tGracias por utilizar nuestro programa" +  "\n\n\tPresione la tecla <Esc> para salir...");  while (Console.ReadKey().Key != ConsoleKey.Escape) ;  }  break;  default:  {  Console.Write("\n\t\t.: OPCIÓN INVALIDA :." +  "\n\n\tPor favor ingrese una opción existente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  }  }  else  {  Console.Write("\n\n\tLa opción no se ha podido capturar, asegúrese de escribirla correctamente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  } while (opcion != '3');  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

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

| Un programa muy similar al anterior, solamente fue necesario hacer las modificaciones necesarias para adecuarlo al ordenamiento de burbuja optimizado y para utilizar strings. No hubo mayor complicación con el programa. |
| --- |