REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 6 |
| Tema a Evaluar | Búsqueda Binaria |
| Fecha de Entrega | 20 de Noviembre del 2023 |

**Programa Fuente**

| using System;  namespace P44\_Busqueda\_Binaria\_1  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio Método Buscar  public static bool Buscar(string[] Arreglo, string elemento, ref int position)  {  int Mitad, Li = 0, Ls = Arreglo.Length - 1;  bool Encontrado = false;  while(Li <= Ls && Encontrado != true)  {  Mitad = (Li + Ls) / 2;  if (Arreglo[Mitad] == elemento) { position = Mitad; Encontrado = true; }  else if (elemento.CompareTo(Arreglo[Mitad]) < 0) Li = Mitad + 1;  else Ls = Mitad - 1;  }  return Encontrado;  }  //Fin Método Buscar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio Método Mostrar  public static void Mostrar(string[] Arreglo)  {  for (int i = 0; i < Arreglo.Length; i++)  {  Console.Write($"\n\t[{i + 1}] - {Arreglo[i]}.");  }  }  //Fin Método Mostrar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio Método Ordenar  public static void IntercalacionSimpleM1(ref string[] ArregloOriginal)  {  //Inicio declaración de variables locales  int i, j, izquierda, derecha, mitad;  string auxiliar;  //Fin declaración de variables locales  //Inicio bloque de instrucciones  for (i = 1; i < ArregloOriginal.Length; i++)  {  auxiliar = ArregloOriginal[i];  izquierda = 0;  derecha = i - 1;  while (izquierda <= derecha)  {  mitad = (izquierda + derecha) / 2;  //auxiliar >= ArregloOriginal[mitad]  if (auxiliar.CompareTo(ArregloOriginal[mitad]) == 1) derecha = mitad - 1;  else izquierda = mitad + 1;  }  j = i - 1;  while (j >= izquierda)  {  ArregloOriginal[j + 1] = ArregloOriginal[j];  j--;  }  ArregloOriginal[izquierda] = auxiliar;  }  }  //Fin Método Ordenar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  static void Main(string[] args)  {  Console.Title = "P44 Búsqueda Binaria 1";  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Declaración de variables  char opcion\_menu;  string nombre\_ciudad;  string[] CiudadesArreglo = new string[15];  int PositionElement = 0;  //Fin Declaración de variables  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Despliegue de menú  do  {  Console.Write("\n\t\t.: MENÚ OPCIONES :." +  "\n\n\t[1] - Insertar Ciudades." +  "\n\n\t[2] - Buscar Ciudades." +  "\n\n\t[3] - Mostrar Ciudades." +  "\n\n\t[4] - Salida del Programa." +  "\n\n\tIngrese el número de la opción deseada: ");  if (char.TryParse(Console.ReadLine(), out opcion\_menu))  {  opcion\_menu = char.ToUpper(opcion\_menu);  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Control de opciones del menú  switch (opcion\_menu)  {  case '1':  {  //Captura de datos  Console.Write("\n\t\t.: INSERTAR CIUDADES :." +  "\n\n\tA continuación se insertarán las 15 ciudades en el arreglo: ");  for (int i = 0; i < CiudadesArreglo.Length; i++)  {  Console.Write($"\n\n\t{i + 1}.- Ingrese la ciudad: ");  CiudadesArreglo[i] = Console.ReadLine();  Console.Write($"\n\tSe ha ingresado la ciudad \"{CiudadesArreglo[i]}\" correctamente");  }  IntercalacionSimpleM1(ref CiudadesArreglo);  Console.Write("\n\n\tSe han ingresado las ciudades correctamente" +  "\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '2':  {  //Captura de datos  Console.Write("\n\t\t.: BUSCAR CIUDADES :." +  "\n\n\tIngrese la ciudad a buscar: ");  nombre\_ciudad = Console.ReadLine();  //Procedimiento de búsqueda  if (Buscar(CiudadesArreglo, nombre\_ciudad, ref PositionElement)) Console.WriteLine($"\n\n\tLa ciudad \"{nombre\_ciudad}\" se encuentra en la posición {PositionElement + 1} de la lista");  else Console.WriteLine($"\n\n\tLa ciudad \"{nombre\_ciudad}\" no se encuentra en la lista");  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '3':  {  //Captura de datos  Console.Write("\n\t\t.: MOSTRAR CIUDADES :." +  "\n\n\tA continuación se mostrarán las 15 ciudades del arreglo: \n");  Mostrar(CiudadesArreglo);  Console.Write("\n\n\tSe han mostrado las ciudades correctamente" +  "\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '4':  {  //Caso 4. Salida del programa  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:  {  //En caso de ingresar una opción inválida  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\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  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();  }  } while (opcion\_menu != '4');  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

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

| Se utilizó parte de la base anterior para construir este programa, pero ahora el método de búsqueda cambió. No se me dificultó nada al pasar el método de pseudocódigo a código C#. Me gustó que se presentara la oportunidad de usar un ordenamiento visto anteriormente y más aún utilizarlo para ordenar strings, no tuve problemas con eso pero me pareció entretenido. |
| --- |