REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 1 |
| Tema a Evaluar | Manejo de funciones matemáticas con estructuras primitivas |
| Fecha de Entrega | 25 de agosto del 2023 |

**Programa Fuente**

| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace P2\_Manejo\_Funciones\_Matematicas\_Estructuras\_Primitivas  {  internal class Program  {  static void Main(string[] args)  {  //Machado Sanchez Javier - 22211600  //Práctica 2 - Manejo de Funciones Matemáticas con Estructuras Primitivas  Console.Title = "Funciones Matemáticas con Estructuras Primitivas";  //Declaración de variables  int primer\_entero, segundo\_entero, exponente;  float primer\_flotante, segundo\_flotante;  //Captura de datos  Console.WriteLine("\n\t\t.: CAPTURA DE ENTEROS :.");  Console.Write("\n\tIngrese el valor del primer entero: ");  primer\_entero = int.Parse(Console.ReadLine());  Console.Write("\n\tIngrese el valor del segundo entero: ");  segundo\_entero = int.Parse(Console.ReadLine());  Console.WriteLine("\n\n\t\t.: CAPTURA DE FLOTANTES :.");  Console.Write("\n\tIngrese el valor del primer flotante: ");  primer\_flotante = float.Parse(Console.ReadLine());  Console.Write("\n\tIngrese el valor del segundo flotante: ");  segundo\_flotante = float.Parse(Console.ReadLine());  Console.WriteLine("\n\n\t\t.: CAPTURA DEL EXPONENTE :.");  Console.Write("\n\tIngrese el valor al que desea elevar los números: ");  exponente = int.Parse(Console.ReadLine());  Console.Write("\n\n\tLos valores han sido capturados correctamente." +  "\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ;  Console.Clear();  //Realización de cálculos e impresión de resultados  Console.WriteLine("\n\t\t.: RESULTADO ENTEROS :.");  Console.WriteLine("\t1 - |{0}| = {1} ..::.. 1.1 - |{2}| = {3} ",primer\_entero,Math.Abs(primer\_entero),segundo\_entero,Math.Abs(segundo\_entero));  //Inicio if's anidados para encontrar raíces negativas  if(primer\_entero > 0 && segundo\_entero > 0)  {  Console.WriteLine("\t2 - V¯{0} = {1} ..::.. 2.1 - V¯{2} = {3}", primer\_entero, Math.Sqrt(primer\_entero), segundo\_entero, Math.Sqrt(segundo\_entero));  }  else if (primer\_entero < 0 && segundo\_entero < 0)  {  Console.WriteLine("\t2 - V¯{0} = No existen raíces negativas ..::.. 2.1 - V¯{1} = No existen raíces negativas", primer\_entero, segundo\_entero);  }  else if(primer\_entero < 0)  {  Console.WriteLine("\t2 - V¯{0} = No existen raíces negativas ..::.. 2.1 - V¯{1} = {2}", primer\_entero, segundo\_entero, Math.Sqrt(segundo\_entero));  }  else if(segundo\_entero < 0)  {  Console.WriteLine("\t2 - V¯{0} = {1} ..::.. 2.1 - V¯{2} = No existen raíces negativas", primer\_entero, Math.Sqrt(primer\_entero), segundo\_entero);  }  //Final if's anidados  Console.WriteLine("\t3 - {0}^{1} = {2} ..::.. 3.1 - {3}^{4} = {5}",primer\_entero,exponente,Math.Pow(primer\_entero,exponente),segundo\_entero, exponente,Math.Pow(segundo\_entero,exponente));  //Inicio if's anidados para encontrar los números mayores y menores  if(primer\_entero == segundo\_entero)  {  Console.WriteLine("\t4 - Ningún número es mayor porque son iguales.");  Console.WriteLine("\t5 - Ningún número es menor porque son iguales.");  }  else if(primer\_entero > segundo\_entero)  {  Console.WriteLine("\t4 - {0} > {1}",primer\_entero,segundo\_entero);  Console.WriteLine("\t5 - {0} < {1}", segundo\_entero, primer\_entero);  }  else  {  Console.WriteLine("\t4 - {0} > {1}", segundo\_entero, primer\_entero);  Console.WriteLine("\t5 - {0} < {1}", primer\_entero, segundo\_entero);  }  //Final if's anidados  Console.WriteLine("\n\t\t.: RESULTADO FLOTANTES :.");  Console.WriteLine("\t1 - |{0}| = {1} ..::.. 1.1 - |{2}| = {3} ", primer\_flotante, Math.Abs(primer\_flotante), segundo\_flotante, Math.Abs(segundo\_flotante));  //Inicio if's anidados para encontrar raíces negativas  if (primer\_flotante > 0 && segundo\_flotante > 0)  {  Console.WriteLine("\t2 - V¯{0} = {1} ..::.. 2.1 - V¯{2} = {3}", primer\_flotante, Math.Sqrt(primer\_flotante), segundo\_flotante, Math.Sqrt(segundo\_flotante));  }  else if (primer\_flotante < 0 && segundo\_flotante < 0)  {  Console.WriteLine("\t2 - V¯{0} = No existen raíces negativas ..::.. 2.1 - V¯{1} = No existen raíces negativas", primer\_flotante, segundo\_flotante);  }  else if (primer\_flotante < 0)  {  Console.WriteLine("\t2 - V¯{0} = No existen raíces negativas ..::.. 2.1 - V¯{1} = {2}", primer\_flotante, segundo\_flotante, Math.Sqrt(segundo\_flotante));  }  else if (segundo\_flotante < 0)  {  Console.WriteLine("\t2 - V¯{0} = {1} ..::.. 2.1 - V¯{2} = No existen raíces negativas", primer\_flotante, Math.Sqrt(primer\_flotante), segundo\_flotante);  }  //Final if's anidados  Console.WriteLine("\t3 - {0}^{1} = {2} ..::.. 3.1 - {3}^{4} = {5}", primer\_flotante, exponente, Math.Pow(primer\_flotante, exponente), segundo\_flotante, exponente, Math.Pow(segundo\_flotante, exponente));  //Inicio if's anidados para encontrar los números mayores y menores  if (primer\_flotante == segundo\_flotante)  {  Console.WriteLine("\t4 - Ningún número es mayor porque son iguales.");  Console.WriteLine("\t5 - Ningún número es menor porque son iguales.");  }  else if (primer\_flotante > segundo\_flotante)  {  Console.WriteLine("\t4 - {0} > {1}", primer\_flotante, segundo\_flotante);  Console.WriteLine("\t5 - {0} < {1}", segundo\_flotante, primer\_flotante);  }  else  {  Console.WriteLine("\t4 - {0} > {1}", segundo\_flotante, primer\_flotante);  Console.WriteLine("\t5 - {0} < {1}", primer\_flotante, segundo\_flotante);  }  //Final if's anidados  //Mensaje final  Console.Write("\n\tLos cálculos han sido realizados correctamente." +  "\n\tPresione la tecla <Esc> para cerrar el programa...");  while (Console.ReadKey().Key != ConsoleKey.Escape) ;  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

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

| Nuevamente la actividad me pareció sencilla, aunque un poco más tardada que la anterior. Al inicio me compliqué un poco tratando de crear un algoritmo para hacer valores absolutos, pero leí que había una función Math y la utilicé. También me pareció un poco tedioso probar todas las combinaciones de números posibles y programar mensajes de error en los casos que fueran necesarios, como en el caso de las raíces negativas. |
| --- |