

Scripting Taller 1

Punto 1: Hacer 5 ejercicios y responder 2 preguntas del archivo:

Punto 9 (Cadena):

Lea una frase de mínimo 3 palabras y convierta cada palabra a minúscula y con mayúscula inicial como muestra el ejemplo:

```
el examen mas facil del mundo  
El Examen Mas Facil Del Mundo
```

Solución:

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        Console.WriteLine("Ingresa una frase de mínimo 3 palabras:");
```

```
        string entrada = Console.ReadLine();
```

```
        string[] palabras = entrada.Split(' ');
```

```
        for (int i = 0; i < palabras.Length; i++)
```

```
        {
```

```
            string palabra = palabras[i];
```

```
            string primeraLetra = palabra.Substring(0, 1);
```

```
            string restoDePalabra = palabra.Substring(1);
```

```
            string nuevaPalabra = primeraLetra.ToUpper() + restoDePalabra.ToLower();
```

```
            Console.Write(nuevaPalabra + " ");
```

```
        }
```

```
    }
```

```
}
```

Punto 14 (Matriz):

Realice una función que lea una matriz de números enteros de NxM y calcule el promedio de cada fila y cada columna.

Solución:

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        Console.WriteLine("Ingresa el número de filas de la matriz:");
```

```
        int filas = int.Parse(Console.ReadLine());
```

```
        Console.WriteLine("Ingresa el número de columnas de la matriz:");
```

```
        int columnas = int.Parse(Console.ReadLine());
```

```
        int[,] matriz = new int[filas, columnas];
```

```
        for (int i = 0; i < filas; i++)
```

```
        {
```

```
            for (int j = 0; j < columnas; j++)
```

```
            {
```

```
                Console.WriteLine($"Ingresa el elemento ({i + 1}, {j + 1}):");
```

```
                matriz[i, j] = int.Parse(Console.ReadLine());
```

```
            }
```

```
        }
```

```
        Console.WriteLine("El promedio de cada fila es:");
```

```
        for (int i = 0; i < filas; i++)
```

```
        {
```

```
            double sumaFila = 0;
```

```
            for (int j = 0; j < columnas; j++)
```

```
            {
```

```
                sumaFila += matriz[i, j];
```

```
            }
```

```
            double promedioFila = sumaFila / columnas;
```

```
            Console.WriteLine($"Fila {i + 1}: {promedioFila}");
```

```

    }

    Console.WriteLine("El promedio de cada columna es:");

    for (int j = 0; j < columnas; j++)
    {
        double sumaColumna = 0;

        for (int i = 0; i < filas; i++)
        {
            sumaColumna += matriz[i, j];
        }

        double promedioColumna = sumaColumna / filas;
        Console.WriteLine($"Columna {j + 1}: {promedioColumna}");
    }
}
}

```

Punto 26:

Usando un ciclo while, ingrese n valores flotantes, cada que ingrese un número acumulelo en una variable, al final calcule el promedio de los números ingresados.

Solución:

```
using System;
```

```
class Program
```

```

{
    static void Main(string[] args)
    {
        Console.WriteLine("Ingresa la cantidad de números a promediar:");
        int n = int.Parse(Console.ReadLine());

        double suma = 0;
        int i = 0;

        while (i < n)
        {
            Console.WriteLine($"Ingresa el número {i + 1}:");
            double numero = double.Parse(Console.ReadLine());

```

```

        suma += numero;
        i++;
    }

    double promedio = suma / n;
    Console.WriteLine($"El promedio de los {n} números ingresados es: {promedio}");
}
}

```

1. Recorra una matriz nxm y reemplace los valores negativos por un número positivo, como muestra el ejemplo:

using System;

```

class Program
{
    static void Main(string[] args)
    {

        // Crear una matriz de tamaño nxm
        int[,] matriz = new int[3, 3] {
            { -1, 2, -3},
            { 6, -8, 9},
            { -3, 4, -5}
        };

        Console.WriteLine("Matriz original:");
        ImprimirMatriz(matriz);

        // Recorrer la matriz y reemplazar los valores negativos
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if (matriz[i, j] < 0)
                {
                    matriz[i, j] = Math.Abs(matriz[i, j]); // Tomar el valor absoluto
                }
            }
        }
    }
}

```

```

    }

    Console.WriteLine("\nMatriz con valores negativos reemplazados:");
    ImprimirMatriz(matriz);
}

static void ImprimirMatriz(int[,] matriz)
{
    int filas = matriz.GetLength(0);
    int columnas = matriz.GetLength(1);

    for (int i = 0; i < filas; i++)
    {
        for (int j = 0; j < columnas; j++)
        {
            Console.Write(matriz[i, j] + "\t");
        }
        Console.WriteLine();
    }
}
}

```

```

Matriz original:
-1      2      -3
6       -8      9
-3      4       -5

Matriz con valores negativos reemplazados:
1       2       3
6       8       9
3       4       5

```

16. Realice la implementación de las firmas de las siguientes invocaciones using System;

```

public class Operaciones
{
    public string Concatenar(params object[] elementos)
    {

```

```

        return string.Concat(elementos);
    }

    public double Fraccionar(string numero)
    {
        if (double.TryParse(numero, out double resultado))
        {
            return resultado / 2.0; // Ejemplo de fraccionar un número: dividir por 2
        }
        else
        {
            throw new ArgumentException("El valor proporcionado no es un número válido.");
        }
    }

    public int Contador(params object[] elementos)
    {
        return elementos.Length;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Operaciones operaciones = new Operaciones();

        // Ejemplo de uso de los métodos
        string concatenacion = operaciones.Concatenar("Hola", " ", "mundo");
        double fraccionado = operaciones.Fraccionar("123.34");
        int contador = operaciones.Contador(1, 3.4, "hola", '3');

        // Imprimir resultados
        Console.WriteLine("Concatenación: " + concatenacion);
        Console.WriteLine("Fraccionado: " + fraccionado);
        Console.WriteLine("Contador: " + contador);
    }
}

```

```

Concatenación: Hola mundo
Fraccionado: 6167
Contador: 4

```

Preguntas:

1. ¿Qué es una variable tipo acumulador? Escriba un ejemplo.

Respuesta:

En programación, un acumulador es una variable que se utiliza para mantener un valor que se actualiza continuamente a medida que se procesa un conjunto de datos. Este valor acumulado puede utilizarse para realizar cálculos, contar elementos o realizar cualquier operación que requiera un seguimiento acumulativo 1.

Por ejemplo, si queremos calcular la suma de los elementos de una lista, podemos utilizar un acumulador para llevar un registro de la suma a medida que recorremos la lista. En el siguiente código en C#, se muestra cómo se puede utilizar un acumulador para calcular la suma de los elementos de una lista:

```
class Program
{
    static void Main(string[] args)
    {
        List<int> numeros = new List<int>() { 1, 2, 3, 4, 5 };
        int suma = 0;

        foreach (int numero in numeros)
        {
            suma += numero;
        }

        Console.WriteLine($"La suma de los elementos de la lista es: {suma}");
    }
}
```

2. ¿Para qué sirve Try Parse?, escriba un ejemplo de cómo se usa?

Respuesta:

El método TryParse se utiliza para convertir una cadena en un tipo de datos específico, como un número entero, flotante o doble. La ventaja de usar TryParse en lugar de Parse es que TryParse no arroja una excepción si la cadena no se puede convertir al tipo de

datos especificado. En su lugar, devuelve un valor booleano que indica si la conversión fue exitosa y almacena el resultado de la conversión en una variable de salida

Aquí un ejemplo de cómo se puede utilizar TryParse en C# para convertir una cadena en un número entero:

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        Console.WriteLine("Ingresa un número entero:");
```

```
        string input = Console.ReadLine();
```

```
        int number;
```

```
        if (int.TryParse(input, out number))
```

```
        {
```

```
            Console.WriteLine($"El número ingresado es: {number}");
```

```
        }
```

```
        else
```

```
        {
```

```
            Console.WriteLine("El valor ingresado no es un número entero válido.");
```

```
        }
```

```
    }
```

```
}
```