



## **INSTITUTO SUPERIOR DE FORMACIÓN DOCENTE Y TÉCNICA N° 93**

**CARRERA:** TECNICATURA SUPERIOR EN ANÁLISIS, DESARROLLO Y PROGRAMACIÓN DE APLICACIONES

**ESPACIO CURRICULAR:** Programación I

**CURSO:** 1 AÑO

**CICLO LECTIVO:** 2023

**PROFESOR/A:** Pini Alejandro

**PLAN AUTORIZADO POR RESOLUCIÓN N°6175/03**

### **Trabajo Práctico N° 1**

**ALUMNA/O:** Sarmiento Alejo

**CANTIDAD DE HOJAS:** #16

**FECHA DE ENTREGA:** 02/11/2023

# Ejercicio 1

---

Código:

```
using System;

// 1 - Desarrollar un algoritmo que cargue una matriz de 4 x 4 con valores enteros
// aleatorios(random)
// comprendidos entre 1 y 6.Luego de cargadas la misma, el programa debe hacer lo
// siguiente:
// a - Imprimir la matriz en consola.
// b - Recorrerla posición a posición, ir sumando el valor de cada celda y luego
// de terminar de
// recorrerla imprimir en consola, y debajo de la matriz ya impresa, el resultado
// de esa suma de
// todos los valores

namespace practical_1
{
    class Program1
    {
        static void Main(string[] args)
        {
            int total_rows = 4; int total_cols = 4; int start = 1; int limit = 7;
            int[,] random_matrix = GetRandomMatrix(total_rows, total_cols, start,
limit);
            PrintAndSumMatrix(random_matrix);
            Console.WriteLine("Presione una tecla para continuar...");
            Console.ReadKey();
        }

        static int[,] GetRandomMatrix(int total_rows, int total_cols, int start,
int limit)
        {
            int[,] matrix = new int[total_rows, total_cols];
            for (int row = 0; row != total_rows; row += 1) {
                for (int col = 0; col != total_cols; col += 1) {
                    matrix[row, col] = new Random().Next(start, limit);
                }
            }
            return matrix;
        }

        static void PrintAndSumMatrix(int[,] matrix) {
            int total_rows = matrix.GetLength(0);
            int total_cols = matrix.GetLength(1);
            int sum = 0;
            Console.WriteLine("La matriz generada es la siguiente:\n");
            for (int row = 0; row != total_rows; row += 1) {
                Console.Write("\t");
```

```

        for (int col = 0; col != total_cols; col += 1) {
            sum += matrix[row, col];
            Console.Write(matrix[row, col] + " ");
        }
        Console.WriteLine();
    }
    Console.WriteLine("\nEl total de la suma entre todos los números de la
matriz es: " + sum + "\n");
}
}
}

```

Salida:

La matriz generada es la siguiente:

```

2 2 3 2
3 6 1 1
6 2 3 3
4 2 2 5

```

El total de la suma entre todos los números de la matriz es: 47

Presione una tecla para continuar...

## Ejercicio 2

---

Código:

```

using System;

// Generar un algoritmo que cargue una matriz de 5 x 5 (con el formato y valores
de la imagen que se muestra
// a continuación mediante un algoritmo automático) y que al final de cada fila
muestre la sumatoria de los
// valores que forman las mismas.

namespace practical_2
{
    class Program
    {
        static void Main(string[] args)
        {
            int total_rows = 5;
            int total_cols = 5;

```

```

        int start = 0;
        int[,] generated_matrix = GenerateMatrix(total_rows, total_cols,
start);
        PrintMatrix(generated_matrix);
        Console.WriteLine("Presione una tecla para continuar...");
        Console.ReadKey();
    }

    static int[,] GenerateMatrix(int total_rows, int total_cols, int start)
    {
        int value = start;
        int[,] matrix = new int[total_rows, total_cols];
        for (int row = 0; row != total_rows; row += 1)
        {
            for (int col = 0; col != total_cols; col += 1)
            {
                matrix[row, col] = value;
                value += 1;
            }
            start += 1;
            value = start;
        }
        return matrix;
    }

    static void PrintMatrix(int[,] matrix) {
        int total_rows = matrix.GetLength(0);
        int total_cols = matrix.GetLength(1);
        int sum = 0;
        Console.WriteLine();
        for (int row = 0; row != total_rows; row += 1)
        {
            Console.Write("\t");
            for (int col = 0; col != total_cols; col += 1)
            {
                sum += matrix[row, col];
                Console.Write(matrix[row, col] + " ");
                if (col == total_cols - 1) {
                    Console.Write("= " + sum);
                    sum = 0;
                }
            }
            Console.WriteLine();
        }
        Console.WriteLine();
    }
}
}
}

```

Salida:

```
0 1 2 3 4 = 10
1 2 3 4 5 = 15
2 3 4 5 6 = 20
3 4 5 6 7 = 25
4 5 6 7 8 = 30
```

Presione una tecla para continuar...

## Ejercicio 3

---

Código:

```
using System;

// Elaborar un programa que lea o recorra una matriz, que ya debe estar cargadas
// con valores, y que imprima
// por pantalla poniendo las filas como columnas y las columnas como filas (lo que
// se conoce como matriz
// transpuesta)

namespace practical_3
{
    class Program
    {
        static void Main(string[] args)
        {
            int[,] matrix = {
                { 2, 5, 6, 7, 8 },
                { 8, 6, 3, 2, 1 },
                { 7, 4, 6, 1, 5 }
            };
            Console.WriteLine("Matrix original:");
            PrintMatrix(matrix);
            int[,] transposed_matrix = TransposeMatrix(matrix);
            Console.WriteLine("Matrix transpuesta:");
            PrintMatrix(transposed_matrix);
            Console.WriteLine("Presione una tecla para continuar...");
            Console.ReadKey();
        }

        static int[,] TransposeMatrix(int[,] old_matrix) {
            int rows = old_matrix.GetLength(0);
            int cols = old_matrix.GetLength(1);
            int[,] new_matrix = new int[cols, rows];
            for (int row = 0; row != rows; row += 1) {
                for (int col = 0; col != cols; col += 1) {
```

```

        new_matrix[col, row] = old_matrix[row, col];
    }
}
return new_matrix;
}

static void PrintMatrix(int[,] matrix)
{
    int total_rows = matrix.GetLength(0);
    int total_cols = matrix.GetLength(1);
    Console.WriteLine();
    for (int row = 0; row != total_rows; row += 1)
    {
        Console.Write("\t");
        for (int col = 0; col != total_cols; col += 1)
        {
            Console.Write(matrix[row, col] + " ");
        }
        Console.WriteLine();
    }
    Console.WriteLine();
}
}
}

```

Salida:

Matrix original:

```

2 5 6 7 8
8 6 3 2 1
7 4 6 1 5

```

Matrix transpuesta:

```

2 8 7
5 6 4
6 3 6
7 2 1
8 1 5

```

Presione una tecla para continuar...

## Ejercicio 4

---

Código:

```

using System;

// Elabore un programa que cargue una matriz de 6 x 6 con números aleatorios entre
// 2 y 8. Imprima la misma
// por pantalla. Al mismo tiempo que la imprime debe almacenar en un matrix de 6
// posiciones los valores de
// la diagonal de dicha matriz (diagonal de izquierda a derecha y de arriba hacia
// abajo) y luego se debe
// imprimir dicho matrix.

namespace practical_4
{
    class Program
    {
        static void Main(string[] args)
        {
            int total_rows = 6;
            int total_cols = 6;
            int start = 2;
            int limit = 9;
            int[,] random_matrix = GetRandommatrix(total_rows, total_cols, start,
limit);

            int[] diagonal_values = GetDiagonalMatrixValues(random_matrix);
            Console.WriteLine("Matriz generada con valores aleatorios:");
            PrintMatrix(random_matrix);
            Console.WriteLine("Array generada con los valores de la diagonal:");
            PrintArray(diagonal_values);
            Console.WriteLine("Presione una tecla para continuar...");
            Console.ReadKey();
        }

        static int[,] GetRandommatrix(int total_rows, int total_cols, int start,
int limit)
        {
            int[,] matrix = new int[total_rows, total_cols];
            for (int row = 0; row != total_rows; row += 1)
            {
                for (int col = 0; col != total_cols; col += 1)
                {
                    matrix[row, col] = new Random().Next(start, limit);
                }
            }
            return matrix;
        }

        static void PrintMatrix(int[,] matrix)
        {
            int total_rows = matrix.GetLength(0);
            int total_cols = matrix.GetLength(1);
            Console.WriteLine();
            for (int row = 0; row != total_rows; row += 1)

```

```

        {
            Console.Write("\t");
            for (int col = 0; col != total_cols; col += 1)
            {
                Console.Write(matrix[row, col] + " ");
            }
            Console.WriteLine();
        }
        Console.WriteLine();
    }

    static int[] GetDiagonalMatrixValues(int[,] matrix)
    {
        int total_rows = matrix.GetLength(0);
        int total_cols = matrix.GetLength(1);
        int[] array = new int[total_cols];
        int index = 0;
        for (int row = 0; row != total_rows; row += 1)
        {
            array[index] = matrix[row, index];
            index += 1;
        }
        return array;
    }

    static void PrintArray(int[] array)
    {
        int array_length = array.GetLength(0);
        Console.WriteLine();
        Console.Write("\t");
        for (int index = 0; index != array_length; index += 1) {
            Console.Write(array[index] + " ");
        }
        Console.WriteLine();
        Console.WriteLine();
    }
}

```

Salida:

Matriz generada con valores aleatorios:

```

5 7 8 4 7 6
3 5 7 4 8 6
4 6 7 5 2 8
7 8 8 3 4 6
7 5 2 3 6 5
2 5 2 7 4 8

```

Array generada con los valores de la diagonal:



5 5 7 3 6 8

Presione una tecla para continuar...

## Ejercicio 5

---

Código:

```
using System;
using System.IO;
using System.Diagnostics;

// 5 - Desarrolle un programa que:
//   a. Utilice dos arrays unidimensionales de 5 posiciones.
//   b. Los arrays deberán ser cargados con valores random de entre 1 y 9.
//   c. Luego deberá tomar cada valor del array1 e ir multiplicando por cada uno
//   de los del array2 de la
//   siguiente forma: 7 x 7 x 9 x 3 x 8 x 9 = 95256

// 8 - A los programas de los ejercicios 5 y 8 deberán incorporarle que además de
//   imprimir por consola imprima
//   también en un archivo de texto plano (.txt) agregando al principio la fecha y
//   hora exacta en que se ejecuta
//   el programa y que sea ese archivo un histórico de ejecuciones (una deba de la
//   otra con su respectiva fecha
//   y hora).

namespace practical_5
{
    class Program
    {
        //////////////////////////////////// WRITER
        ////////////////////////////////////
        static string file_name = "practical_5_output.txt";
        // Combino el path del escritorio y el nombre del archivo. Obtengo la ruta
        // del escritorio con Environment.
        static string file_path = Path.Combine(
            path1: Environment.GetFolderPath(Environment.SpecialFolder.Desktop),
            path2: file_name
        );
        static StreamWriter writer = File.AppendText(file_path);

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

```

        writer.WriteLine($"\\n{DateTime.Now.ToString()}\\n");
        int array_length = 6;
        int rndm_start = 1;
        int rndm_limit = 10;
        int[] array_1 = GenerateRandomArray(array_length, rndm_start,
rndm_limit);
        int[] array_2 = GenerateRandomArray(array_length, rndm_start,
rndm_limit);
        int[] results = LMultiply(array_1, array_2);
        WriteLine("Array 1:");
        PrintArray(array_1);
        WriteLine("Array 2:");
        PrintArray(array_2);
        WriteLine("Resultante:");
        PrintArray(results);
        Console.WriteLine("Presione una tecla para continuar...");
        Console.ReadKey();
        writer.Close();
    }

    static int[] GenerateRandomArray(int array_length, int rndm_start, int
rndm_limit)
    {
        int[] array = new int[array_length];
        for (int index = 0; index != array_length; index += 1)
        {
            array[index] = new Random().Next(rndm_start, rndm_limit);
        }
        return array;
    }

    static void PrintArray(int[] array)
    {
        int array_length = array.GetLength(0);
        WriteLine();
        Write("\\t");
        for (int index = 0; index != array_length; index += 1)
        {
            Write(array[index] + " ");
        }
        WriteLine();
        WriteLine();
    }

    static int[] LMultiply(int[] array_1, int[] array_2) {
        int pos = 0;
        int[] results = new int[array_1.Length];
        for (int i = 0; i != array_1.Length; i += 1) {
            results[i] = array_1[i] * array_2[i];
            if (i != array_1.Length - 1) pos = i + 1;
            else pos = 0;
            while (true) {
                if (pos == i) break;
                results[i] *= array_2[pos];
            }
        }
    }

```

```

        if (pos == (array_2.Length - 1)) {
            pos = 0;
            continue;
        }
        pos++;
    }
}
return results;
}

static void Write(string text = "")
{
    Console.Write(text);
    writer.Write(text);
}

static void WriteLine(string text = "")
{
    Console.WriteLine(text);
    writer.WriteLine(text);
}
}
}

```

Salida:

15/9/2023 21:45:14

Array 1:

5 4 2 5 9 3

Array 2:

5 5 1 7 1 9

Resultante:

7875 6300 3150 7875 14175 4725

15/9/2023 21:45:18

Array 1:

8 1 5 7 8 8

Array 2:

3 7 2 1 1 9

Resultante:

3024 378 1890 2646 3024 3024

15/9/2023 21:45:22

Array 1:

4 1 9 8 8 6

Array 2:

4 8 2 8 8 1

Resultante:

16384 4096 36864 32768 32768 24576

## Ejercicio 6

---

Código:

```
using System;
using System.Diagnostics;

// Desarrollar un programa que utilice un array bidimensional que permita cargar
// Nombre, Apellido y Edad de
// 5 personas (y en ese orden). Luego de que el mismo es cargado con la ayuda o
// interacción del usuario, el
// programa deberá recorrer el array e imprimir por consola el contenido de este
// respetando el formato que
// se muestra en la siguiente imagen ejemplo de ejecución

namespace practical_6
{
    class Program
    {
        static void Main(string[] args)
        {
            int ammount = 5;
            string[,] matrix = CreateUserMatrix(ammount);
            PrintMatrix(matrix);
            Console.WriteLine("Presione una tecla para continuar...");
            Console.ReadKey();
        }
    }
}
```

```

    }

    static string[,] CreateUserMatrix(int ammount)
    {
        string[,] matrix = new string[ammount, 3];
        for (int row = 0; row != ammount; row++) {
            Console.Write("Ingrese el nombre: ");
            matrix[row, 0] = Console.ReadLine();
            Console.Write("Ingrese el apellido: ");
            matrix[row, 1] = Console.ReadLine();
            Console.Write("Ingrese la edad: ");
            matrix[row, 2] = Console.ReadLine();
        }
        Console.WriteLine();
        matrix = OrderMatrixByAge(matrix);
        return matrix;
    }

    static string[,] OrderMatrixByAge(string[,] matrix)
    {
        int total_rows = matrix.GetLength(0);
        int total_cols = matrix.GetLength(1);
        string[,] new_matrix = new string[total_rows, total_cols];
        List<int> order = new List<int>();
        for (int row = 0; row != total_rows; row++)
        {
            // Si el valor del indice actual es mayor al valor del indice
            anterior
            // hago sobre la lista order un .Add, sino un hago un .Insert
            cuando corresponda.
            if (order.Count == 0 || int.Parse(matrix[row, 2]) >
            int.Parse(matrix[order[order.Count - 1], 2]))
                order.Add(row);
            else
            {
                for (int insert_index = 0; insert_index < order.Count();
            insert_index++)
                {
                    if (int.Parse(matrix[row, 2]) <
            int.Parse(matrix[order[insert_index], 2]))
                    {
                        order.Insert(insert_index, row);
                        break;
                    }
                }
            }
        }
        // Creo la nueva matriz en base al la lista order que contiene los
        indices en orden de menor a mayor.
        for (int row = 0; row != total_rows; row++)
        {
            new_matrix[row, 0] = matrix[order[row], 0];
            new_matrix[row, 1] = matrix[order[row], 1];
            new_matrix[row, 2] = matrix[order[row], 2];
        }
    }
}

```

```

    }
    return new_matrix;
}

static void PrintMatrix(string[,] matrix)
{
    int total_rows = matrix.GetLength(0);
    for (int row = 0; row != total_rows; row++)
        Console.WriteLine(
            $"{matrix[row, 1]}, {matrix[row, 0]} {matrix[row, 2]}"
        );
    Console.WriteLine();
}
}
}

```

Salida:

```

Ingrese el nombre: Alejo
Ingrese el apellido: Sarmiento
Ingrese la edad: 22
Ingrese el nombre: Pepe
Ingrese el apellido: Perez
Ingrese la edad: 12
Ingrese el nombre: Maria
Ingrese el apellido: Marta
Ingrese la edad: 55
Ingrese el nombre: Leandro
Ingrese el apellido: Pini
Ingrese la edad: 43
Ingrese el nombre: Charly
Ingrese el apellido: Garcia
Ingrese la edad: 71

Perez, Pepe 12
Sarmiento, Alejo 22
Pini, Leandro 43
Marta, Maria 55
Garcia, Charly 71

Presione una tecla para continuar...

```

## Ejercicio 7

---

Código:

```

using System;
using System.IO;

// 7 - Desarrollar un programa que simule que se arrojan 5 dados, 5 veces, sobre
// la mesa y se toma nota de los
// valores de cada dado (la cara que queda para arriba). Los valores deben ser
// cargados en un array
// bidimensional siendo cada fila la que representa cada una de las 5 tiradas o
// tiros.
// Luego de estar cargado, el programa deberá recorrer el array e indicar
// cuantas veces salió cada cara del
// dado en total de todos los tiros.

namespace practical_7
{
    class Program
    {
        //////////////////////////////////// WRITER
        ////////////////////////////////////
        static string file_name = "practical_7_output.txt";
        // Combino el path del escritorio y el nombre del archivo. Obtengo la ruta
        // del escritorio con Environment.
        static string file_path = Path.Combine(
            path1: Environment.GetFolderPath(Environment.SpecialFolder.Desktop),
            path2: file_name
        );
        static StreamWriter writer = File.AppendText(file_path);

        ////////////////////////////////////
        ////////////////////////////////////
        static void Main(string[] args)
        {
            writer.WriteLine($"\\n{DateTime.Now.ToString()}\\n");
            int rolls = 5;
            int dices = 5;
            int faces = 6;
            int[,] results = StartGame(rolls, dices, faces);
            ShowResults(results);
            Console.WriteLine("Presione una tecla para continuar...");
            Console.ReadKey();
            writer.Close();
        }

        static int[,] StartGame(int rolls, int dices, int faces)
        {
            int[,] results = new int[rolls, dices];
            faces = faces + 1;
            Random rndm = new Random();
            for (int i = 0; i != rolls; i++)
            {

```

```

        WriteLine($"Presione una tecla para la jugada numero {i + 1}");
        Console.ReadKey();
        for (int j = 0; j != dices; j++)
        {
            results[i, j] = rndm.Next(1, faces);
        }
    }
    WriteLine();
    return results;
}

static void ShowResults(int[,] results)
{
    int[,] occurrences = new int[results.GetLength(0) + 1, 2];
    for (int i = 0; i != results.GetLength(0); i++)
    {
        Write($"Tiro {i + 1}: ");
        for (int j = 0; j != results.GetLength(1); j++)
        {
            Write($"{results[i, j]} ");
            occurrences[results[i, j] - 1, 0] = results[i, j];
            occurrences[results[i, j] - 1, 1] = occurrences[results[i, j]
- 1, 1] + 1;
        }
        WriteLine("\n");
    }
    for (int i = 0; i != occurrences.GetLength(0); i++)
    {
        WriteLine($"{occurrences[i, 0]} = {occurrences[i, 1]} veces");
    }
}

static void Write(string text = "")
{
    Console.Write(text);
    writer.Write(text);
}

static void WriteLine(string text = "")
{
    Console.WriteLine(text);
    writer.WriteLine(text);
}
}
}

```

Salida:

15/9/2023 21:43:24



Presione una tecla para la jugada numero 1  
Presione una tecla para la jugada numero 2  
Presione una tecla para la jugada numero 3  
Presione una tecla para la jugada numero 4  
Presione una tecla para la jugada numero 5

Tiro 1: 1 6 2 2 4

Tiro 2: 4 3 4 2 1

Tiro 3: 3 6 6 3 6

Tiro 4: 2 1 3 6 1

Tiro 5: 4 5 6 5 3

1 = 4 veces

2 = 4 veces

3 = 5 veces

4 = 4 veces

5 = 2 veces

6 = 6 veces

15/9/2023 21:43:29

Presione una tecla para la jugada numero 1  
Presione una tecla para la jugada numero 2  
Presione una tecla para la jugada numero 3  
Presione una tecla para la jugada numero 4  
Presione una tecla para la jugada numero 5

Tiro 1: 4 2 5 2 3

Tiro 2: 5 5 1 4 2

Tiro 3: 1 4 2 4 2

Tiro 4: 1 4 5 1 3

Tiro 5: 2 4 6 3 3

1 = 4 veces

2 = 6 veces

3 = 4 veces

4 = 6 veces

5 = 4 veces

6 = 1 veces

15/9/2023 21:43:34

Presione una tecla para la jugada numero 1  
Presione una tecla para la jugada numero 2  
Presione una tecla para la jugada numero 3  
Presione una tecla para la jugada numero 4

Presione una tecla para la jugada numero 5

Tiro 1: 4 6 3 3 3

Tiro 2: 3 1 5 2 3

Tiro 3: 3 3 5 2 2

Tiro 4: 2 5 4 3 3

Tiro 5: 6 1 1 4 3

1 = 3 veces

2 = 4 veces

3 = 10 veces

4 = 3 veces

5 = 3 veces

6 = 2 veces

15/9/2023 21:43:38

Presione una tecla para la jugada numero 1

Presione una tecla para la jugada numero 2

Presione una tecla para la jugada numero 3

Presione una tecla para la jugada numero 4

Presione una tecla para la jugada numero 5

Tiro 1: 4 1 3 3 5

Tiro 2: 6 4 3 3 6

Tiro 3: 5 2 4 4 3

Tiro 4: 1 4 3 5 6

Tiro 5: 3 6 1 4 4

1 = 3 veces

2 = 1 veces

3 = 7 veces

4 = 7 veces

5 = 3 veces

6 = 4 veces