

Ejercicios Matrices

Índice

1. [Ejercicio 1](#)
2. [Ejercicio 2](#)
3. [Ejercicio 3](#)
4. ☒ [Ejercicio 4](#)
5. [Ejercicio 5](#)
6. [Ejercicio 6](#)
7. ☒ [Ejercicio 7](#)
8. ☒ [Ejercicio 8](#)
9. [Ejercicio 9](#)

Ejercicio 1

Crea un programa que cree un array bi-dimensional de **10 x 10**.

Inicializa la matriz de forma que las **filas pares se rellenen con unos y las impares a ceros**.

Una vez inicializada la matriz, muestra su contenido en pantalla.

Ejercicio 2

Crea un programa que cree un array bi-dimensional de **5 x 5**.

Inicializa la matriz de forma que los componentes pertenecientes a la **diagonal** de la matriz tomen valor **uno**, y el resto valor cero. Una vez inicializada la matriz.

Muestra su contenido en pantalla con un **foreach**.

Ejercicio 3

Dada una matriz de 3x5, diseña un programa que lea dicha matriz y posteriormente cree una nueva matriz a partir de la primera **permutando filas por columnas**. A esta nueva matriz la llamaremos matriz traspuesta.

$$A_{2 \times 3} = \begin{bmatrix} -1 & 2 & 1 \\ -3 & 3 & 5 \end{bmatrix} \Rightarrow A_{2 \times 3}^T = \begin{bmatrix} -1 & 3 \\ 2 & 5 \\ 1 & 5 \end{bmatrix}$$

✓ Ejercicio 4

Tenemos una **tabla dentada de caracteres** que tiene almacenados los nombres de **20 países**. Se pide diseñar un programa que realice, tantas veces como sea requerido por el usuario, las siguientes operaciones:

1. Buscar un país.
2. Pedir prefijo telefónico.
3. Añadir prefijo al nombre del país.

Nota: Dejaremos un espacio en blanco entre el nombre y el código telefónico.

Otras consideraciones...

- El prefijo estará formado por 2 caracteres.
- Cuando se acabe el bucle, el programa debe presentar en pantalla todos los países en orden alfabético.

Nota: Para ordenar utilizaremos una función con el método de ordenación de la burbuja que nos ordene el array.

Ejercicio 5

Dada la siguiente tabla dentada:

```
int[][] m = new int[][]  
{  
    new int [] {1,2,3,4},  
    new int [] {5,6,7},  
    new int [] {9,10,11,12,5},  
    new int [] {9,10}  
};
```

Haz un programa que busque en la tabla y posteriormente muestre, el **array con mas número de elementos**, usando solo bucles foreach.

Nota: No hace falta que crees ningún método auxiliar.

Ejercicio 6

Crea un programa para **mostrar** los elementos de la siguiente tabla dentada:

```
int[][][] tabla = new int[][][]  
{  
    new int [][]  
    {  
        new int [] {1,2,3,4},  
        new int [] {5,6,7},  
        new int [] {9,10,11,12}  
    },  
    new int [][]  
    {  
        new int [] {13,14,15,16},  
        new int [] {17,18,19,20},  
        new int [] {21,22}  
    },  
};
```

Cada tabla se debe mostrar de la siguiente manera.

Utiliza las funciones necesarias para que el código quede claro.

1,2,3,4	13,14,15,16
5,6,7	17,18,19,20
9,10,11,12	21,22

✓ Ejercicio 7

Escribe un programa que se encargue de controlar el **aforo de un Multicine**:

- El **cine** tendrá **tres salas** (A, B, C), en las cuales se pasarán diariamente tres sesiones (1ª, 2ª, 3ª).
- El número **máximo de personas** de cada una de las salas es:
 - Sala **A** = **200** personas.
 - Sala **B** = **150** personas.
 - Sala **C** = **125** personas.
- Tendremos un menú con dos opciones:
 1. Venta de entradas.
 2. Estadística de aforo.
- Para salir del programa se tendrá que pulsar la tecla **ESC**.

Cada vez que se realice una venta de entradas se pedirá:

- El número de entradas que se van a comprar.
- La sala
- La sesión a la que se quiere asistir.

Las entradas vendidas quedarán registradas en el array.

Si el número de entradas sobrepasa el aforo máximo de la sala, se indicará mediante un mensaje por pantalla.

En la opción de estadística de aforo, se mostrará una tabla de la siguiente manera:

	Sesión1	Sesión2	Sesión3
SalaA	178	100	99
SalaB	12	50	100
SalaC	32	101	55

✓ Ejercicio 8

Realiza un programa para **crear una tabla dentada de enteros**.

- Para cada ello deberás crear una array de dos elementos, donde cada elemento será una tabla dentada de enteros.
- El programa pedirá cuantas filas tiene y el número de elementos de cada fila, para cada una de las tablas dentadas.
- Reservará la memoria para cada una y rellenará la fila con los elementos generados de forma aleatoria.

Posteriormente se mostrará el array de dos matrices dentadas de la siguiente forma:

1, 2, 3, 4	13, 14, 15, 16
5, 6, 7	17, 18, 19, 20
9, 10, 11, 12	21, 22

Nota: Recuerda que las tablas se dimensionan con la palabra reservada new.

Ejercicio 9

Dado el siguiente código:

```
static void Main()
{
    int[,] m1 = new int[,] { { 1, 3, 5 }, { 2, 4, 6 } };
    int[,] m2 = new int[,] { { 6, 5 }, { 4, 3 }, { 2, 1 } };

    int[,] m3 = MultiplicaMatricesEnteras(m1, m2);
}
```

Implementa el código del método estático:

```
static int[,] MultiplicaMatricesEnteras(int[,] m1, int[,] m2);
```

Para multiplicar dos matrices debes tener en cuenta lo siguiente:

$$\begin{bmatrix} A \end{bmatrix}_{m,n} \cdot \begin{bmatrix} B \end{bmatrix}_{n,p} = \begin{bmatrix} C \end{bmatrix}_{m,p}$$

*Dada una matriz **A** de **m** filas y **n** columnas y una matriz **B** de **n** filas y **p** columnas, el número de columnas de la primera debe coincidir con el número de filas de la segunda.*

Nota: Si las matrices de entrada no cumplen la condición para poder ser multiplicadas, se mostrará un mensaje de error informativo.

El resultado será una matriz C de m filas y p columnas (las filas de A y las columnas de B) de tal manera que C_{ij} (fila i-ésima y columna j-ésima de C) se obtiene de sumar los productos de los elementos correspondientes a la fila i-ésima de la primera matriz (A), por los de la columna j-ésima de la segunda matriz (B).

Ejemplo resultado de multiplicar las matrices m1 y m2 de código del main:

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 6 & 5 \\ 4 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1*6 + 3*4 + 5*2 & 1*5 + 3*3 + 5*1 \\ 2*6 + 4*4 + 6*2 & 2*5 + 4*3 + 6*1 \end{bmatrix} = \begin{bmatrix} 28 & 19 \\ 40 & 28 \end{bmatrix}$$

Nota: Para solucionar el ejercicio debes tener en cuenta que el método `m1.GetLength(dimension)` devuelve el número de elementos de `m1` en la dimensión especificada, de tal manera que:

`m1.GetLength(0)` me devuelve el número de filas de `m1` y `m1.GetLength(1)` me devuelve el número de columnas.