

# Ejercicios Adicionales de Cadenas

---

1. Escribe la función **EscribeNombres** a la que le pasamos una cadena por parámetro y nos la escribe por la pantalla (sí, con `Console.WriteLine`) con el siguiente formato: la cadena contendrá varias palabras separadas por espacios, con el siguiente formato: "apellido1 apellido2 nombre apellido1 apellido2 nombre (...)" y deberemos escribirla por pantalla de la siguiente forma:

```
nombre apellido1 apellido2
nombre apellido1 apellido2
(...)
```

2. Escribe la función **CuentaLetras** a la que le pasas una cadena por parámetro y te devuelve un array de enteros. El tamaño del array será 5 y el contenido será el número de veces que aparecen las letras "a", "b", "c", "d" y "e", respectivamente, en la cadena que le hemos pasado (independientemente de si son mayúsculas o minúsculas).

Ej.: cadena = "patata camaleón batata", resultado: [8, 1, 1, 0, 1]

**Versión Pro:** La función devolverá un array de enteros con 26 posiciones correspondientes a las 26 letras del abecedario. No uséis un *switch* (ni 26 *ifs*). Perdón, 27, se me olvidaba la eñe.

3. Escribe la función **MayusculasMinusculas** que recibirá dos parámetros: una cadena (que podrá contener saltos de línea) y un número entre 1 y 4. Dependiendo del número que le pasemos, la función nos devolverá otra cadena con las siguientes características:

- 1 = Todas las letras en minúsculas.
- 2 = Todas las letras en mayúsculas.
- 3 = La primera letra de cada palabra en mayúsculas y el resto en minúsculas.
- 4 = La primera letra de cada frase en mayúsculas y el resto en minúsculas.

Escribir también un pequeño menú en el programa principal para probar las diferentes opciones.

4. Escribe la función **EliminaTags** a la que le pasamos una cadena que contiene tags (como los de XML) y los elimina dejando sólo el texto (nos devuelve una cadena con el resultado).

Ej.: `<p>Esto es texto normal <b>y esto es texto en negrita</b>.</p>`  
Nos devolvería: `Esto es texto normal y esto es texto en negrita.`

5. Escribe la función **OrdenaPalabrasComas** a la que le pasamos una cadena que contiene varias palabras separadas por comas y nos devuelve otra cadena con las mismas palabras separadas por comas, pero ordenadas en orden alfabético.

Ej.: "hola,don,pepito,pasó,usted,ya,por,casa"

Devolvería: "casa,don,hola,pasó,pepito,por,usted,ya"

6. Escribe la función **AcentosHTML** que te sustituye los caracteres acentuados que le paséis por el código HTML correspondiente. La función recibirá una cadena por parámetro y nos devolverá otra cadena con el resultado. Aquí tenéis una tabla con las equivalencias:

signo	mnemo
<	&lt;
>	&gt;
&	&amp;
"	&quot;
á	&aacute;
Á	&Aacute;
é	&eacute;
É	&Eacute;
í	&iacute;

signo	mnemo
í	&Iacute;
ó	&oacute;
Ó	&Oacute;
ú	&uacute;
Ú	&Uacute;
ñ	&ntilde;
Ñ	&Ntilde;
ü	&uuml;
Ü	&Uuml;

7. Escribe la función **PalabrasMismaLetra** a la que le pasamos una cadena y nos devolverá un array de cadenas que contendrá las palabras que empiezan y acaban por la misma letra. A la hora de contar las palabras, habrá que ignorar los símbolos de puntuación y las mayúsculas y minúsculas.

Ej.: "Amanda tiene tres serpientes." devolvería {amanda, serpientes}

8. Escribe una función **NumeroTexto99** a la que le pasamos un entero y nos devuelve una cadena con ese número puesto como texto (p.ej.: 76 = "setenta y seis").
- El número deberá estar comprendido entre 0 y 99. En caso contrario, devuelve una cadena vacía.
  - Se recomienda escribir la función NumeroTexto9 que hace lo mismo, pero sólo con números de 1 cifra, y usarla para simplificar esta función.
  - Si me ponéis un switch de 100 elementos os echo de clase directamente.
  - Si no sabéis como se escriben los números del uno al noventa y nueve, os mando de vuelta a la ESO.
  - OPCIONAL: Escribir la función NumeroTexto999.