Ejercicios Adicionales de Cadenas

1. Escribe una función **MarcaSubCadena** a la que le pasamos dos cadenas y nos busca la segunda cadena dentro de la primera. Nos devolverá otra cadena igual que la primera, pero en la que se cambiará la primera letra de cada vez que aparece la subcadena por un asterisco.

2. Escribe la función **EscribeNombres** a la que le pasamos una cadena por parámetro y nos la escribe por la pantalla (con *println*) 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
   (…)
   ```

3. 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 *if*s). Perdón, 27, se me olvidaba la eñe.

4. 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.

5. 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.`

6. 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`"

7. 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
<	<
>	>
&	&
"	"
á	á
Á	Á
é	&aecute;
É	É
í	í

signo	mnemo
Í	Í
ó	ó
Ó	Ó
ú	ú
Ú	Ú
ñ	ñ
Ñ	Ñ
ü	ü
Ü	Ü

8. 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`}

9. 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.

10. Escribe la función **LimpiaCadena** a la que le pasamos una cadena que incluirá letras y signos de puntuación. La función quitará todos los símbolos de la cadena, dejando tan solo un espacio entre cada palabra, y devolverá la cadena resultante.
    Ejemplo: Si le pasamos "Hola, ¿qué tal? Soy amigo de Poti-Poti." Nos devolvería "Hola qué tal Soy amigo de Poti Poti".

11. Escribe la función **CompruebaEmail**, a la que le pasamos un string con una dirección de correo electrónico y nos devuelve true si es válida y false si no lo es.

Para que una dirección de correo sea correcta se tienen que dar las siguientes condiciones:

- Tiene que tener una arroba '@' y solamente una. Además, la arroba no podrá estar ni al principio ni al final (la arroba divide las dos partes de la dirección).
- Cada una de las dos partes de la dirección de email se compondrá de los siguientes caracteres:
  - Letras minúsculas y mayúsculas (sin acentos y sin la eñe)
  - Números
  - El guion '-', aunque no podrá estar ni al principio ni al final de cada parte.
  - El punto '.', que no podrá estar ni al principio ni al final de cada parte ni haber dos puntos seguidos.

Ejemplos: very.common@example.com, admin@mailserver1

12. Escribe la función **PalabrasImpares** a la que le pasamos una cadena de caracteres y nos devuelve otra cadena de caracteres con el resultado. La función contará el número de letras de cada palabra y dejará dentro de la cadena sólo las palabras con un número de letras impar.

    Ej.: "El perro de San Roque" -> "perro San Roque"

13. Escribe la función **CuentaDiptongos** a la que le pasamos una cadena de caracteres y nos devuelve un entero, que será el número de diptongos que hay en la cadena. Para los que se saltaron la ESO, un diptongo es cuando hay dos vocales seguidas, siempre y cuando las dos vocales no sean fuertes (o sea, "ae", "ea", "ao", "oa", "eo" y "oe" no son diptongos).

    Ej.: "Puede caer una bien buena" -> 3 diptongos

    Para que el ejercicio esté completo, se han de tener en cuenta también:
    - Los acentos: comió -> sí, oído -> no
    - La semivocal 'y': voy -> sí, yo -> no

    Opcional: para subir nota, podéis intentar hacerlo con estas excepciones:
    - La 'u' muda: queso -> no, guepardo -> no
    - La 'u' con diéresis: cigüeña -> sí
    - La 'h' intercalada: ahijado -> sí

14. Escribe la función **FiltraArrayPalabras**, a la que le pasamos un array de cadenas de caracteres (que generalmente será resultante de hacer un split) y nos devolverá otro array de cadenas en el que sólo aparecerán las cadenas que estén compuestas íntegramente por letras. Es decir, si una cadena de nuestro array contiene números, signos de puntuación u otros símbolos, no aparecerá en el array que devolvemos.
    Ej.: Si le pasamos el array: ["patata", "c3po", "5874", "tomate", "pimiento?"],
    nos devolverá: ["patata", "tomate"]