1) Invertir cadena

Crea una clase *CadenaUtils* dentro de la cual exista un método llamado *invertir*, el cual reciba una cadena de texto y devuelva esa misma cadena, pero al revés.

Crea una clase Principal para probar la funcionalidad creada.
Ejemplo: invertir("azul") devuelve "luza"

2) Concatenar cadenas con operador +

Dentro de la clase *CadenaUtils,* crear un método llamado *concatenar* cuya función sea pasados parámetros de tipo String, cadena1 y cadena2 que devuelva el resultado de concatenar ambas cadenas usando el operador +.

Crea una clase Principal para probar la funcionalidad creada.
Ejemplo: concatenar(*"Mi planeta es la tierra.", "Y es azul") devuelve "Mi planeta es la tierra.Y es azul"*

Además, en esta clase Principal, mostrar en pantalla la siguiente información del objeto String resultante teniendo en cuenta el ejemplo anterior:

- Su longitud.
- El carácter asociado a la posición 7 y a la posición 22.
- La subcadena "planeta".
- La posición que ocupa el carácter 'Y'.
- La cadena transformada en mayúsculas.
- Por último, comprobar si el primer carácter de la cadena es 'M' y mostrar por consola un mensaje que lo indique.

3) Comparar cadenas

Dentro de la clase *CadenaUtils,* crear un método llamado *comparar* e implementar la siguiente funcionalidad. Leer dos palabras por teclado y que nos muestre en pantalla:

- si tienen la misma longitud o no. En caso de que NO la tengan, escribir cuál de ellas tiene más caracteres.

- si son exactamente iguales o no lo son.

- si son iguales sin tener en cuenta mayúsculas y minúsculas.

Crea una clase Principal para probar la funcionalidad creada. Leer las palabras desde el programa Principal y pasarlas por parámetro al método comparar.

Salida esperada por consola:

Tecleado abeto y aBeto
La longitud es la misma
NO son exactamente iguales
pero sí lo son si no tenemos en cuenta mayúsculas y minúsculas

4) Partir cadenas

Dentro de la clase *CadenaUtils,* crear un método llamado *dividir* e implementar la siguiente funcionalidad. Leer una cadena por teclado y la "rompa" en dos mitades. Después deberá escribir en pantalla cada una de las mitades.

A la hora de "romper" la cadena tener en cuenta:

- Las mitades serán iguales siempre que la cadena tenga un número de caracteres par.

- En caso de que el número de caracteres sea impar no se podrá hacer la mitad exacta, pero partiremos la cadena lo más aproximado a la mitad.

Crea una clase Principal para probar la funcionalidad creada. Desde el programa principal pasar la cadena que debe romperse por parámetro al método dividir.

Salida esperada por consola:

Tecleado: abetillo - Saldrá por consola: abet | illo

Tecleado: abeto - Saldrá por consola: abe | to

5) Búsqueda de cadenas

Dentro de la clase *CadenaUtils,* crear un método llamado *buscarLetra* e implementar la siguiente funcionalidad. Pasar por parámetro la letra a buscar y la cadena de caracteres sobre la que realizar la búsqueda. El método debe mostrar por consola el número de veces que aparece la letra a buscar en la cadena pasada por parámetro.

Además, si el número de veces que se repite la letra es superior a 5, debe aparecer el mensaje "**Exceso de** " y la letra correspondiente.

Crea una clase Principal para probar la funcionalidad creada.

6) Palindromo

Dentro de la clase *CadenaUtils,* crear un método llamado *palindromo* que indique si la palabra pasada por parámetro es o no un palíndromo. Un palíndromo es una cadena que se lee igual de izquierda a derecha que de derecha a izquierda.

Primero será necesario ignorar los espacios en blanco. Utiliza un método de la clase String para eliminar dichos espacios.
Crea una clase Principal para probar la funcionalidad creada.

Salida esperada por consola:

Tecleado: dabale arroz a la zorra el abad
Traza: dabalearrozalazorraelabad → Resultado: SI es palíndromo

7) Calcular Edad

Dentro de la clase *CadenaUtils*, crear un método llamado *calcularEdad* que realice la siguiente funcionalidad. Dados por parámetro tu año de nacimiento y el año actual. La aplicación deberá devolver una cadena (objeto String) con el valor de tu edad y después añadirle la palabra "años". Escribe en pantalla mensajes de traza con el contenido de la cadena que has creado.

Crea una clase Principal para probar la funcionalidad creada. Desde la clase Principal lee por teclado tu año de nacimiento y el año actual y pásaselo por parámetro al método creado.

NOTA: Recuerda que el método genérico **valueOf** convierte valores a String.

Usa el método **concat** en lugar del + para concatenar cadenas.

8) Secuencia de Números

Dentro de la clase *CadenaUtils*, crear un método llamado *secuenciaNumeros* que dado un número por parámetro, debe realizar la siguiente funcionalidad. Dado un número entero, debe devolver un objeto String con todos los dígitos desde 0 al número pasado por parámetro.

- Por ejemplo, para 5, la cadena devuelta tendrá
  "0 1 2 3 4 5"
- Por ejemplo, para 12, la cadena devuelta tendrá
  "0 1 2 3 4 5 6 7 8 9 10 11 12"

Prueba el método haciendo dos o tres llamadas con números diferentes, desde la clase Principal.

NOTA: Recuerda la diferencia entra los objetos **String** y **StringBuffer / StringBuilder**.

9) PalabraMayorApp

Crea la Aplicación **es.maestre.str. PalabraMayorApp** para que lea por teclado tres palabras por teclado e indique cual de las 3 palabras es la mayor. Implenta la funcionalidad en la clase **es.maestre.str.Palabras** donde crees el método **getPalabraMayor** que, dadas dos cadenas, me devuelva la cadena mayor. En caso de que sean iguales, debe devolver un **null**.

Nota: Utiliza el método del API **compareTo**

10) NumBaseApp

Crea una clase Java llamada **es.maestre.Conversor** que contenga solo los siguientes métodos **:**

a) **decToBin**

Este método tomará un número en decimal (en base 10) y devolverá una cadena con el número equivalente en binario (en base 2).

En la cabecera del método escribe los modificadores **public** y **static**:

public static String **decToBin** (int valor)

> **AVISO:**
> Recuerda que debes utilizar la clase **StringBuffer** por eficiencia para ir concatenando los dígitos binarios.

b) **binToDec**

Este método tomará un número en binario (cadena con ceros y unos) y devolverá el número en decimal.

En la cabecera del método escribe los modificadores **public** y **static**:

Crea ahora una **aplicación Java** llamada NumBaseApp que lea un número por teclado en decimal y escriba en pantalla el número en binario.

- Deberás utilizar el método **decToBin** que creaste anteriormente.

- Haz además algunas llamadas al método **binToDec** con distintos valores para comprobar que el resultado que devuelve el método es correcto.

11) Primera letra

Escribir un método que dada una cadena de caracteres devuelva la primera letra de cada palabra. Por ejemplo, si recibe 'Universal Serial Bus' debe devolver 'USB'

12) Vocales

Escribir un método que devuelva solo las vocales. Por ejemplo, si recibe 'algoritmos' debe devolver 'aoio'.

13) Suma de Números

Escribir un método que devuelve la suma de los caracteres correspondientes a dígitos decimales pertenecientes a la secuencia almacenada en un String. Por ejemplo, si la cadena es ABC12m4XYZ entonces la rutina debe devolver el valor numérico entero 7.

14) Signo Zodiacal

Crea un programa donde calcules el signo zodiacal del usuario. El programa pedirá el día y mes de nacimiento del usuario y tendrá que llamar a un método que calcula el signo zodiacal pasado el día y mes de nacimiento. El método que calcule el signo zodiacal devolverá una cadena de caracteres similar a:

*El **"día"** de **"mes"** pertenece a **"signoZodical"***

Ten en cuenta la siguiente información de los signos zodiacales:

♈ 21/03 - 20/04 **Aries**	♉ 21/04 - 21/05 **Tauro**	♊ 22/05 - 21/06 **Géminis**	♋ 22/06 - 22/07 **Cáncer**
♌ 23/07 - 22/08 **Leo**	♍ 23/08 - 22/09 **Virgo**	♎ 23/09 - 22/10 **Libra**	♏ 23/10 - 22/11 **Escorpio**
♐ 23/11 - 21/12 **Sagitario**	♑ 22/12 - 20/01 **Capricornio**	♒ 21/01 - 19/02 **Acuario**	♓ 20/02 - 20/03 **Piscis**

15) Basket

Realizar un programa en Java en el que simule el resultado de un partido de baloncesto.

Para ello el usuario tendrá que pedir el nombre de los dos equipos y que el programa los lea y que cree un resultado aleatorio para cada cuarto, este número aleatorio serán los puntos que el equipo ha metido en ese cuarto y varía de 10 a 30.

La puntuación de cada uno de los equipos será la suma de los puntos de los 4 cuartos que tiene el partido y una vez obtenido el resultado final, el programa deberá comparar ambos marcadores e imprimir quien ha ganado de la siguiente forma: **Ha ganado el equipo** *X* **al equipo** *Y* **por** *Num* **a** *Num*

Siendo un ejemplo: **Ha ganado el Real Madrid al Barca por 76 a 68**