Ejercicios Unidad 5: Cadenas de texto

1.- Conteste con *verdadero* o *falso* a cada una de las siguientes proposiciones; en caso de ser *falso*, explique por qué.

a) Cuando los objetos String se comparan utilizando ==, el resultado es true si los objetos String contiene los mismos valores.
b) Un objeto String puede modificarse una vez creado.

2.- Para cada uno de los siguientes enunciados, escriba una instrucción que realice la tarea indicada:

a) Comparar la cadena en s1 con la cadena en s2 para ver si su contenido es igual.
b) Anexar la cadena s2 a la cadena s1, utilizando +=.
c) Determinar la longitud de la cadena en s1.

3.- Escriba una aplicación que utilice el método compareTo de la clase String para comparar dos cadenas introducidas por el usuario. Muestre si la primera cadena es menor, igual o mayor que la segunda.

4.- Escriba una aplicación que reciba como entrada una línea de texto y que la imprima dos veces; una vez en letras mayúsculas y otra en letras minúsculas.

5.- Escriba una aplicación que reciba como entrada una línea de texto y un carácter de búsqueda, y que utilice el método indexOf de la clase String para determinar el número de ocurrencias de ese carácter en el texto.

6.- Escriba una aplicación con base en el programa del ejercicio anterior, que reciba como entrada una línea de texto y utilice el método indexOf de la clase String para determinar el número total de ocurrencias de cada letra del alfabeto en ese texto. Las letras mayúsculas y minúsculas deben contarse como una sola. Imprima los valores en formato tabular.

7.- Escriba una aplicación que lea una línea de texto y que imprima sólo aquellas palabras que comiencen con la letra "b".

8.- Escriba una aplicación que lea una línea de texto y que imprima sólo aquellas palabras que comiencen con las letras "ED".

9.- Escriba una aplicación que reciba como entrada un código entero para un carácter y que muestre el carácter correspondiente.

10.- Modifique la aplicación anterior de manera que genere todos los posibles códigos de tres dígitos en el rango de 000 a 255, y que intente imprimir los caracteres correspondientes.

11.- Escriba sus propias versiones de los métodos de búsqueda indexOf y lastIndexOf de la clase String.

12.- Escriba una aplicación que lea una línea de texto desde el teclado e imprima una tabla que indique el número de ocurrencias de cada letra del alfabeto en el texto. Por ejemplo, la frase:

Ser o no ser: ése es el dilema:

contiene una “a”, ninguna “b”, ninguna “c”, etcétera.

13.- Escriba una aplicación que lea una línea de texto e imprima una tabla que indique el número de palabras de una letra, de dos letras, de tres letras, etcétera, que aparezcan en el texto. Por ejemplo, en la siguiente tabla se muestra la cuenta para la frase:

¿Qué es más noble para el espíritu?

Longitud de palabra	***Ocurrencias***
1	0
2	2
3	2
4	1
5	1
6	0
7	0
8	1

14.- *(Impresión de fechas en varios formatos)* Las fechas se imprimen en varios formatos comunes. Dos de los formatos más utilizados son:

25/05/2020 y 25 de abril de 2020

Escriba una aplicación que lea una fecha en el primer formato e imprima dicha fecha en el segundo formato.

15.- *(Protección de cheques)* Para evitar la alteración de una cantidad en un cheque, la mayoría de los sistemas computarizados que emiten cheques emplean una técnica llamada protección de cheques. Los cheques diseñados para impresión por computadora contienen un número fijo de espacios en los cuales la computadora puede imprimir una cantidad. Suponga que un cheque contiene ocho espacios en blanco en los cuales la computadora puede imprimir la cantidad de un cheque de nómina. Si la cantidad es grande, entonces se llenarán los ocho espacios. Por ejemplo:

```
1,230.60      (cantidad del cheque)
--------
12345678      (números de posición)
```

Por otra parte, si la cantidad es menor de $1,000, entonces varios espacios quedarían vacíos. Por ejemplo:

```
   99.87
--------
12345678
```

contiene tres espacios en blanco. Si se imprime un cheque con espacios en blanco, es más fácil para alguien alterar la cantidad del cheque. Para evitar que se altere el cheque, muchos sistemas de escritura de cheques insertan *asteriscos al principio* para proteger la cantidad, como se muestra a continuación:

```
***99.87
--------
12345678
```

Escriba una aplicación que reciba como entrada una cantidad a imprimir sobre un cheque y que lo escriba mediante el formato de protección de cheques, con asteriscos al principio si es necesario. Suponga que existen nueve espacios disponibles para imprimir la cantidad.

16.- *(Clave Morse)* Quizá el más famoso de todos los esquemas de codificación es el código Morse, desarrollado por Samuel Morse en 1832 para usarlo con el sistema telegráfico. El código Morse asigna una serie de puntos y guiones a cada letra del alfabeto, cada dígito y algunos caracteres especiales (tales como el punto, la coma, los dos puntos y el punto y coma). En los sistemas orientados a sonidos, el punto representa un sonido corto y el guión representa un sonido largo. Otras representaciones de puntos y guiones se utilizan en los sistemas orientados a luces y sistemas de señalización con banderas. La separación entre palabras se indica mediante un espacio o, simplemente, con la ausencia de un punto o un guión. En un sistema orientado a sonidos, un espacio se indica por un tiempo breve durante el cual no se transmite sonido alguno.

Escriba una aplicación que lea una frase en español y que codifique la frase en clave Morse. Además, escriba una aplicación que lea una frase en código Morse y que la convierta en su equivalente en español. Use un espacio en blanco entre cada letra en clave Morse, y tres espacios en blanco entre cada palabra en clave Morse.

Carácter	***Código***	***Carácter***	***Código***	***Carácter***	***Código***	***Carácter***	***Código***
A	. -	J	. - - -	S	. . .	1	. . - - -
B	- . . .	K	- . -	T	-	2	. . . - -
C	- . - .	L	. - . .	U	. . -	3	. . . . -
D	- . .	M	- -	V	. . . -	4	. . . . .
E	.	N	- .	W	. - -	5	- . . . .
F	. . - .	O	- - -	X	- . . -	6	- - . . .
G	- - .	P	. - - .	Y	- . - -	7	- - - . .
H	. . . .	Q	- - . -	Z	- - . .	8	- - - - .
I	. .	R	. - .	0	. - - - -	9	- - - - -