16) Código Morse

Crea un programa que sea capaz de transformar texto natural a código morse.

- En morse se soporta raya "—", punto ".", un espacio " " entre letras o símbolos y dos espacios entre palabras " ".
- A continuación, se muestra la equivalencia de cada letra y número en código morse
- Usa constantes para definir la equivalencia de 26 letras y 10 números del código morse.

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

17) Conversor de Temperatura

Crea un método que transforme grados Celsius en Fahrenheit y viceversa.

- Para que un dato de entrada sea correcto debe poseer un símbolo "°" y su unidad ("C" o "F"). En caso contrario retornará un error.
- Usa la siguiente tabla para saber la equivalencia entre grados Celsius y Fahrenheit

Pasar de grados Celsius (°C) a Fahrenheit (°F)	Pasar de grados Fahrenheit (°F) a Celsius (°C)
°F = (°C · 1,8) + 32	°C = (°F - 32) / 1,8

18) Comparador de Fechas

Crea un método que calcule y retorne cuántos días hay entre dos cadenas de texto que representen fechas.

- Una cadena de texto que representa una fecha tiene el formato "dd/MM/yyyy". El método recibirá dos String y retornará un int.
- Comprueba que la primera cadena represente una fecha mayor que la segunda cadena. En caso de no serlo, lanzar una excepción
- Si una de las dos cadenas de texto no representa una fecha correcta se lanzará una excepción
- Para lanzar una excepción nueva utiliza: `throw new Exception("Mensaje error";`

19) Comparando caracteres

Crea un método que reciba dos cadenas como parámetro (str1, str2) e imprima otras dos cadenas como salida (out1, out2).

- out1 contendrá todos los caracteres presentes en la str1 pero NO estén presentes en str2.
- out2 contendrá todos los caracteres presentes en la str2 pero NO estén presentes en str1.
- Puedes usar el método *contains* de *String*, que devuelve true si y solo si, la cadena contiene la secuencia de caracteres especificada.

20) La encriptación de Karaca

Crea un método que sea capaz de encriptar texto utilizando el algoritmo de encriptación de Karaca. Sigue los siguientes pasos para hacerlo:

1. Solo se devolverán palabras en minúsculas, por lo que el primer paso es pasar el texto a minúsculas
2. Invierte la palabra que se quiere encriptar y ha sido pasada como parámetro. Ejemplo “apple” será “elppa”
3. Cambia todas las vocales siguiendo la siguiente regla:

a => 0, e => 1, i => 2, o => 3, u => 4

“elppa” será ahora “1lppO”

4. Añade “aca” al final de la palabra “1lppO” será ahora “1lppOaca”

Según estos pasos para la palabra “apple” deberá devolver “1lppOaca”.

21) Vocal más común

Crea un método que reciba un texto y retorne la vocal que más veces se repita. Si no hay vocales podrá devolver vacío.

22) Juego del ahorcado

Realiza un programa para jugar al juego del ahorcado, donde el ordenador debe elegir una palabra aleatoriamente de entre las que tenga almacenadas en un array y ofrecer al usuario tantos huecos como caracteres tenga dicha palabra.

El usuario dispondrá de un número limitado de intentos para adivinar la palabra, pudiendo indicar un carácter en cada intento. Si el carácter existe en la palabra se mostrará en la pantalla y el usuario habrá consumido el intento.

Para la parte de elegir la palabra a adivinar. Definir las palabras a adivinar, según el siguiente ejemplo. Después elegir un número aleatorio entre 0 y el número de palabras que guardéis en el array (para el ejemplo mostrador, será entre 0 y 2) y elegir así la palabra del array que se deberá adivinar. Por ejemplo, si el numero aleatorio es 1, seleccionar arrayPalabras[1].

```
String arrayPalabras[] = new String[3];
arrayPalabras[0] = "hola";
arrayPalabras[1] = "adios";
arrayPalabras[2] = "ejemplo";
```

23) Comprobar DNI

Realiza una aplicación que compruebe si un número de DNI (con la letra) es correcto, para lo cual se deben seguir las siguientes reglas:

- Un DNI está formado por 9 caracteres alfanuméricos, donde los 8 primeros caracteres han de ser números y el último debe ser una letra.
- La letra del DNI se calcula dividiendo el valor del número de 8 dígitos entre 23 y el resto nos indica que letra le corresponde siguiendo la siguiente tabla.

Tabla para calcular la letra del DNI

RESTO	0	1	2	3	4	5	6	7	8	9	10	11
LETRA	T	R	W	A	G	M	Y	F	P	D	X	B

RESTO	12	13	14	15	16	17	18	19	20	21	22
LETRA	N	J	Z	S	Q	V	H	L	C	K	E

24) Contraseña

Haz una clase llamada *Password* que siga las siguientes condiciones:

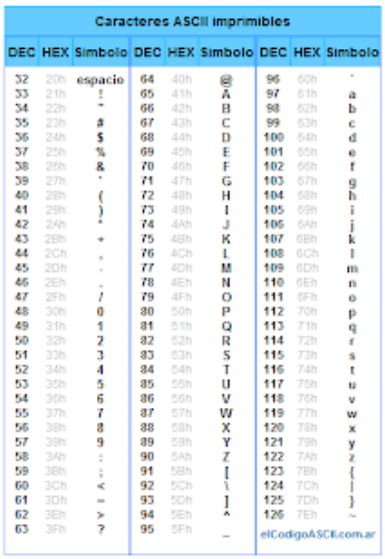
- Que tenga los atributos longitud y contraseña. Por defecto, la longitud será de 8.

Los constructores serán los siguientes:

- Un constructor por defecto.
- Un constructor con la longitud que nosotros le pasemos. Generará una contraseña aleatoria con esa longitud.

Los métodos que implementa serán:

- esFuerte(): devuelve un booleano si es fuerte o no, para que sea fuerte debe tener al menos 2 mayúsculas, 1 minúscula y 5 números.
- generarPassword(): genera la contraseña del objeto con la longitud que tenga. En Java, podemos convertir un número entero a su correspondiente carácter ASCII usando el método Character.toString(int). Fíjate en la tabla adjunta, para ver la equivalencia entre números en decimal y el correspondiente símbolo ASCII.
- Método get para contraseña y longitud.
- Método set para longitud.



25) Lotería

Realizar un programa que simule la creación de apuestas de tres tipos diferentes (Euromillón, Primitiva y Quinigoles).

En primer lugar, se deberá preguntar el tipo de apuesta que quiere realizar (Euromillón, Primitiva y Quinigoles), después crear la apuesta elegida y por último imprimir la apuesta generada mostrándola por consola.

Para generar la apuesta de la primitiva, se deberán obtener 6 números aleatorios del 1 al 49 y un reintegro aleatorio del 0 al 9.

Para generar la apuesta de euromillones, se deberán obtener 5 números aleatorios del 1 a 50 y dos estrellas aleatorias del 1 al 11.

Para Quinigoles, el usuario deberá indicar el número de goles que se marcarán en uno de los partidos de la jornada, por ejemplo: Barcelona – Real Sociedad.

Crea las clases necesarias para representar toda la información, sin usar arrays ni listas.

La impresión de los boletos queda a vuestra elección. Por ejemplo, para imprimir la apuesta de la primitiva, crear un método al que como parámetro le entre un objeto Primitiva que devuelva un String con toda la información, por ejemplo:

***EUROMILLONES**: 7 - 33 - 46 – 11 – 24 / Estrellas: 5 - 3*