

### Práctica 1: Conversión

La gravedad de la Luna es aproximadamente el 17% de la gravedad de la Tierra. Se pide:

1. Crear un programa que, dado un peso de la Tierra, calcule su equivalente en la Luna.
2. El programa debe mostrar por pantalla un mensaje similar a este: "Un peso de 30 kilos en la Tierra equivalen a X kilos en la Luna".

### Práctica 2: Operaciones matemáticas

Crear un programa que realice algunas operaciones matemáticas.

1. Pedir por consola al usuario que introduzca 3 números enteros y almacenar su valor en variables.
2. Operación 1.  $a*b/c$  Mostrar por consola el resultado
3. Operación 2.  $(a*c)\%b$  Mostrar por consola el resultado
4. Operación 3.  $2*(a+c-b)/(b*c)$  Mostrar por consola el resultado
5. Operación 4.  $((a*c)+(b*a))/a-c$  Mostrar por consola el resultado
6. Opcional: Repetir las operaciones con números decimales para ver las diferencias de resultado.

### Práctica 3: Notas

Crear un programa que indique al usuario la nota media global de su curso. Requisitos:

1. El programa pedirá al usuario que introduzca 5 notas, para las asignaturas: Matemáticas, Física, Química, Lenguaje e Historia.
2. El programa realizará la media de esas cinco notas.
3. El programa indicará al usuario el rango al que equivale la nota final que ha sacado:
  - a. Entre 0 – 3 Muy deficiente
  - b. Entre 3 – 5 Insuficiente
  - c. Entre 5 – 6 Suficiente
  - d. Entre 6 – 7 Bien
  - e. Entre 7 – 9 Notable
  - f. Entre 9 – 10 Sobresaliente

### Práctica 4: Contar espacios

Crear un programa que lea caracteres desde teclado indefinidamente. Requisitos:

1. El programa pedirá al usuario que inserte caracteres o frases completas por teclado de manera indefinida
2. El programa contará el número de espacios que se van introduciendo por consola
3. El programa finalizará cuando el usuario introduzca un punto "."
4. El programa imprimirá por consola el número total de espacios que se han introducido

### Práctica 5: Armstrong

Desarrollar un programa que determine si un número es un número de Armstrong. Un número de Armstrong es aquel que es igual a la suma de sus dígitos elevados a la potencia de su número de cifras.

Recursos para el ejercicio:

- `Math.floor`: Devuelve el máximo entero menor o igual a un número pasado como parámetro.
- `Math.log10`: Devuelve el logaritmo en base 10 de un número pasado como parámetro.
- `Math.pow`: Devuelve el valor del primer argumento elevado a la potencia del segundo argumento.

### Práctica 6: Conversión 2

Al igual que se hizo con la Práctica 1 de conversión de pesos, en este caso se pide crear un programa que imprima una tabla de conversión de pulgadas a metros. Requisitos:

1. Un metros son 39,37 pulgadas
2. Elaborar el programa desde la pulgada 1 hasta la 144
3. La tabla debe de dejar un espacio libre cada 12 pulgadas para ser más legible.

### Práctica 7: CandyCalculator

En las fiestas de un barrio hay competiciones deportivas que premian el desempeño en ellas con cupones que luego puedes cambiar por golosinas. Una barra de caramelo se puede cambiar por 10 cupones, y un chicle por 3 cupones.

Escribe una clase `CandyCalculator` que tenga un método `candyCalculator` que permita:

1. Calcular, dado un número dado de cupones, cuantas barras de caramelo y chicles puedes obtener si gastas todos tus cupones en barras de caramelo primero, y utilizas los cupones restantes en chicles.
2. También te tiene que devolver el número de cupones restantes que no puedes gastar.
3. Devuelve los valores en un array en donde la posición [0] representa las barras de caramelo, la posición [1] los chicles y la posición [2] los cupones restantes.

### Práctica 8: Elementos Duplicados

Crear un programa que, dado un array de números enteros, determine cuales son sus elementos que se encuentran duplicados. Por ejemplo:

```
int [] arrayDePrueba = {1, 2, 3, 3, 9, 8, 7, 4, 6, 7, 0, 4, 5};
```

### Práctica 9: Contar vocales

Implementa la función `int contarVocales(String)` que dada una cadena, cuenta el número de vocales que existe en la misma. No importa que las vocales estén en mayúsculas o minúsculas y el resto de caracteres que no sean vocales son ignorados. También se ignoran las vocales acentuadas.

### **Práctica 10: Persona**

Se requiere un programa que modele el concepto de una persona. Una persona posee nombre, apellido, número de documento de identidad y año de nacimiento. La clase debe tener un constructor que inicialice los valores de sus respectivos atributos.

La clase debe incluir los siguientes métodos:

1. Definir un método que imprima por pantalla los valores de los atributos del objeto.
2. En el método main se deben crear dos personas y mostrar los valores de sus atributos por pantalla.

### **Práctica 11: Rectángulo**

Crea una clase Rectángulo con las siguientes características

- Incluyen un constructor al que se le pasan los datos de base y altura. Si se intenta dar valor negativo a alguna de las dimensiones, lo corrige al valor positivo usando `Math.abs(int a)`.
- Un constructor sin parámetros que inicializa un nuevo rectángulo con base 2 y altura 1.
- Un método `esCuadrado` que indica si el rectángulo actual es un cuadrado.
- Un método `area` que calcula el área del rectángulo actual.
- Un método `perimetro` que calcula el perímetro del rectángulo actual.
- Un método `gira` que gira 90 grados el rectángulo actual (intercambiado la base por la altura).

### **Práctica 12: Bag**

Crea una clase Bag que nos permita almacenar una colección desordenada de objetos en la que, al extraerlos, todos tengan la misma probabilidad de ser escogidos.

En el archivo adjunto tenéis el código con los métodos que hay que crear y una explicación de cómo deben funcionar.