MOOC Init. Prog. C++
Ejercicios semana 3

Ejercicio 7: tablas de multiplicar (iteración `para`, nivel 1)

> Este ejercicio corresponde al ejercicio 3 (páginas 19 y 199) del libro *C++ par la pratique* (3ª edición, PPUR).

Objetivo

Escribe un programa `tables.cc` para mostrar las tablas de multiplicar del 2 al 10. Su programa debe producir la siguiente salida:

```
Tablas de multiplicar

Tabla de 2 :
  1 * 2 = 2
  ...
  10 * 2 = 20
...
Tabla de 5 :
  1 * 5 = 5
  2 * 5 = 10
  ...
...
Tabla de 10 :
  1 * 10 = 10
  ...
```

Método :

Utiliza dos estructuras de iteración `for` anidadas.

Ejercicio 8: Pelotas que rebotan (iteración `para`, nivel 2)

> Este ejercicio corresponde al ejercicio 6 (páginas 21 y 202) del libro *C++ par la pratique* (3ª edición, PPUR).

Objetivo:

El objetivo de este ejercicio es resolver el siguiente problema:

Cuando una pelota cae desde una altura inicial *h*, su velocidad al llegar al suelo es $v = \sqrt{2 \times h \times g}$. Inmediatamente después del rebote, su velocidad es $v1 = eps \times v$ (donde *eps* es una constante y *v* es la velocidad antes del rebote). A continuación, se eleva hasta la altura $h = \frac{{v_1}^2}{2g}$.

Se trata de escribir un programa (`rebonds1.cc`) que calcule la altura a la que sube la pelota tras un número `nbr` de rebotes.

Método :

Queremos resolver este problema, no desde un punto de vista formal (ecuaciones), sino mediante **simulación.**
el sistema físico (la pelota).

Utilice a `para la` iteración y las variables `v`, `v1`, (las velocidades antes y después del rebote), y `h`, `h1`
(alturas al inicio de la caída y al final del ascenso).

Tareas :

Escribe el programa `rebonds1.cc` que muestre la altura tras el número de rebotes especificado.

Su programa debe utilizar la **constante** `g`, con un valor de 9,81, y pedir al usuario que introduzca los valores de

- **H0** (altura inicial, restricción: H0 > 0),
- **eps** (coeficiente de rebote, restricción 0 <= eps < 1)
- **nbr** (número de rebotes, restricción: 0 <= NBR).

Pruebe los valores H0 = 25, eps = 0,9, NBR = 10. La altura obtenida debe ser aproximadamente 3,04.

Nota:

- Para utilizar funciones matemáticas (como `sqrt()`), añada

Ejercicio 8: Pelotas que rebotan (iteración `para`, nivel 2)

> Este ejercicio corresponde al ejercicio 6 (páginas 21 y 202) del libro *C++ par la pratique* (3ª edición, PPUR).

`#include <cmath>` al principio del archivo fuente.

Ejercicio 9: Pelotas que rebotan - el retorno (bucles do...while, nivel 2)

Este ejercicio corresponde al ejercicio 7 (páginas 22 y 203)
del libro *C++ par la pratique* (3ª edición, PPUR).

Ahora preguntamos cuántos rebotes hace esta pelota antes de que la altura a la que rebota sea inferior (o igual) a una altura dada `h_fin`.

Escribe el programa `rebonds2.cc`, que muestra el número de rebotes en la pantalla.

Debería utilizar un bucle `do...while`, y pedir al usuario que introduzca los valores para :

- **H0** (altura inicial, restricción: H0 > 0),
- **eps** (coeficiente de rebote, restricción 0 <= eps < 1)
- **h_fin** (altura final deseada, restricción: 0 < h_fin < H0).

Prueba H0=10, eps=0.9, h_fin=2. Deberías obtener 8 rebotes.

Ejercicio 10: Números primos (estructuras de control, nivel 2)

> Este ejercicio corresponde al ejercicio 9 (páginas 22 y 205) de *C++ par la pratique* (3ª edición, PPUR).

Escriba el programa `premier.cc` que pide al usuario que introduzca un número entero **n** estrictamente mayor que 1, y luego decide si este número es primo o no.

Algoritmo :

1. Comprueba si el número **n** es par (si lo es, no es primo a menos que sea 2).
2. Para todos los números impares menores o iguales que la raíz cuadrada de **n**, comprueba si dividen a **n**. Si no es así, entonces n es primo. Si no es así, **n** es primo.

Tareas :

- Si **n** no es primo, tu programa mostrará el mensaje: "El número no es primo, porque es divisible por *D*", donde *D* es un divisor de n distinto de 1 y n.
- En caso contrario, aparecerá el mensaje: "Creo firmemente que este número es primo".

Prueba tu programa con los números: 2, 16, 17, 91, 589, 1001, 1009, 1299827 y 2146654199. Indica qué números son primos.

Los resultados deberían ser los siguientes:

```
2 es primo
16 no es primo, porque es divisible por 2
17 es primero
91 no es primo, porque es divisible por 7
589 no es primo porque es divisible por 19 1001
no es primo porque es divisible por 7 1009 es
primo
1299827 es el primero
2146654199 no es primo, porque es divisible por 46327
```
