

# Fundamentos de la programación

## Sesión de Laboratorio 3

1. Queremos diseñar un programa para un cajero automático que calcule la cantidad de billetes y monedas que hay que darle al cliente cuando retira una determinada importe del banco. El cliente solicitará una cantidad en el formato (“euros,centimos”) y el programa calculará el número de billetes y monedas de ditintos valores que tiene que utilizar para sumar esa cantidad, *maximizando el número de billetes/monedas de mayor a menor valor*. Es decir, primero calculará el máximo número de billetes de 500 euros; después, para la cantidad restante, el máximo número de billetes de 200 euros; luego el máximo de billetes de 100 euros, y así sucesivamente hasta llegar a las monedas de céntimo. Una posible ejecución puede ser:

Cuánto dinero quieres? ("euros,centimos"): 1494,69

Tu cambio:

Billetes de 500: 2	Monedas de 2: 2
Billetes de 200: 2	Monedas de 1: 0
Billetes de 100: 0	Monedas de 50c: 1
Billetes de 50: 1	Monedas de 20c: 0
Billetes de 20: 2	Monedas de 10c: 1
Billetes de 10: 0	Monedas de 5c: 1
Billetes de 5: 0	Monedas de 2c: 2
	Monedas de 1c: 0

2. Se dice que un año es *bisiesto* si es divisible entre 4, a menos que sea divisible entre 100. Sin embargo, si un año es divisible entre 100 y además es divisible entre 400, también resulta bisiesto. Escribir un programa que pida un año al usuario y determine si es o no bisiesto (sin utilizar la condicionales *if-else*).
3. Implementar un programa que solicite los coeficientes  $a$ ,  $b$  y  $c$  de una ecuación de segundo grado  $ax^2 + bx + c = 0$  y calcule sus dos soluciones, suponiendo que efectivamente la ecuación tiene soluciones reales.
4. Debido a la escasez de agua se pretende implantar un sistema de tarifas que penalice el consumo excesivo de este recurso, de acuerdo con la siguiente tabla:

Consumo ( $m^3$ )	euros/ $m^3$
Primeros 100	0.15
De 100 a 500	0.20
De 500 a 1000	0.35
A partir de 1000	0.80

Implementar un programa que solicite el consumo de agua en  $m^3$  y calcule la factura de acuerdo con la tabla anterior.

En los siguientes ejercicios utiliza el compilador para experimentar y confirmar tus respuestas.

5. Suponiendo que `n` es de tipo `int`, empareja las descripciones con la definición correspondiente:

- |                                                         |                        |
|---------------------------------------------------------|------------------------|
| ▪ El mayor número par no superior a <code>n</code>      | ▪ $(n / 2) * 2$        |
| ▪ El primer número par mayor o igual que <code>n</code> | ▪ $n + ((n + 1) \% 2)$ |
| ▪ El primer impar mayor o igual que <code>n</code>      | ▪ $((n + 1) / 2) * 2$  |

6. Determinar cuáles de las siguiente expresiones son erróneas y cuáles correctas. En el caso de las expresiones correctas indicar el tipo que deben tener las variables que aparecen, el tipo de la expresión completa y el valor resultante de la evaluación, si es posible calcularlo:

- |                             |                                                   |                                         |
|-----------------------------|---------------------------------------------------|-----------------------------------------|
| ▪ <code>x == y</code>       | ▪ <code>(k % 2 != 0)    ((k + 1) % 2 != 0)</code> | ▪ <code>((n * (n - 1)) % 2 != 0)</code> |
| ▪ <code>p &gt; p + 1</code> | ▪ <code>(2 * 2) / 4 == 2 * (2 / 4)</code>         | ▪ <code>p == q    r</code>              |
| ▪ <code>p == true</code>    | ▪ <code>'b' - 'a' &gt; 0</code>                   | ▪ <code>10 / 3 == 10 / 3.0</code>       |

7. Dadas las variables booleanas `x`, `y`, `z`, con valores `x=true`, `y=false`, `z=true`, evaluar las expresiones que aparecen a continuación:

- |                                                     |                                       |
|-----------------------------------------------------|---------------------------------------|
| ▪ <code>(x &amp;&amp; y)    (x &amp;&amp; z)</code> | ▪ <code>x    y &amp;&amp; z</code>    |
| ▪ <code>(x    !y) &amp;&amp; (!x    z)</code>       | ▪ <code>!(x    y) &amp;&amp; z</code> |

8. Indicar cuáles de las siguientes asignaciones son correctas y determinar el tipo que deben tener las variables que aparecen:

- |                                   |                                                    |
|-----------------------------------|----------------------------------------------------|
| ▪ <code>a = (b == c == d);</code> | ▪ <code>a = 2.3 / b;</code>                        |
| ▪ <code>x = 2 &lt; 1;</code>      | ▪ <code>2 = a;</code>                              |
| ▪ <code>x = x + 1;</code>         | ▪ <code>p = Math.Sqrt(2) == Math.Sqrt(2.0);</code> |
| ▪ <code>c = 2 &lt;= 7;</code>     | ▪ <code>x = 'y';</code>                            |

9. Las siguientes expresiones tienen sentido, pero no son válidas en la sintaxis de C#. Traducirlas a expresiones booleanas válidas.

`x < y <= z`    `a > b > c`    `x` es diferente de `a` y de `b`

Dos cualesquiera de los valores `x`, `y`, `z` son iguales