



## Composición condicional. Años bisiestos

### Problema 1

Escribe un programa completo (declaraciones, especificación a través de una precondition y una postcondition y función main) que indique si un año, previamente solicitado al usuario y leído del teclado, es bisiesto o no. El programa debe presentar una forma de interacción con el usuario como la que se muestra en los dos ejemplos siguientes:

```
Escriba un año: 2016  
El año 2016 es bisiesto
```

```
Escriba un año: 2018  
El año 2018 no es bisiesto
```

(El texto resaltado en negrita y subrayado corresponde con posibles entradas suministradas por el usuario a través del teclado. El resto del texto representan escrituras del programa en la pantalla.)

### Problema 2

Es posible que el programa que hayas planteado en el problema anterior sea incorrecto desde el punto de vista del dominio del problema que estamos resolviendo. Averigua cuándo un año es bisiesto o no.

Indica cuáles de los siguientes años han sido o serán bisiestos:

2000	2008	2016	2018	2019	2020
1900	2100				
1492	1500	1582	1600	1700	

(Quizá hubiera sido conveniente que este problema se hubiera hecho antes que el problema 1. El **análisis** del problema que hay que resolver debe preceder al diseño e implementación de su solución)

### Problema 3

Escribe una solución correcta al problema de saber si un año es bisiesto o no. En esta ocasión, apóyate en una función denominada `esBisiesto` que debes implementar y cuya especificación debe ser la siguiente:

```
/*  
 * Pre:  agno > 1582  
 * Post: Ha devuelto true si y solo si el año «agno» es bisiesto de acuerdo con el calendario  
 *       gregoriano.  
 */  
bool esBisiesto(int agno) {  
    ...  
}
```

Modifica la función main del problema 1 para que haga uso de la función `esBisiesto` que has implementado.

## Composición condicional. Matrículas de honor

### Problema 4

Escribe una función (precondition, postcondition, cabecera y cuerpo) que determine el número de matrículas de honor que pueden otorgarse por grupo de docencia y asignatura en la Universidad de Zaragoza.

El artículo 14.2 del acuerdo de 22 de diciembre de 2010, del Consejo de Gobierno, por el que se aprueba el *Reglamento de normas de evaluación y aprendizaje de la Universidad de Zaragoza* establece:

La mención «Matrícula de Honor» podrá ser otorgada a estudiantes que hayan obtenido una calificación igual o superior a 9 en la asignatura. Su número no podrá exceder del 5% de los estudiantes matriculados en dicha asignatura en el correspondiente grupo docente, salvo cuando sea inferior a 20, en cuyo caso se podrá conceder una sola matrícula de honor.



## Composición iterativa. Tablas de multiplicar

### Problema 5

Escribe un programa completo que solicite al usuario reiteradamente por el valor de una tabla de multiplicar que desee que sea escrita por el programa y que, a continuación, el programa la escriba, salvo cuando la respuesta del usuario sea 0, en cuyo caso el programa concluirá. El programa debe presentar una forma de interacción con el usuario como la que se muestra en el ejemplo siguiente:

```
¿Qué tabla desea escribir? (0 para acabar): 7
```

```
LA TABLA DEL 7
```

```
7 x 0 = 0
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

```
¿Qué tabla desea escribir? (0 para acabar): 6
```

```
LA TABLA DEL 6
```

```
6 x 0 = 0
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
...
6 x 9 = 54
6 x 10 = 60
```

```
¿Qué tabla desea escribir? (0 para acabar): 0
```

Sería altamente recomendable que, en la resolución de este problema, se reutilice la función `presentarTabla` que se utilizó como ejemplo en clase de teoría y cuya especificación aparece al final del enunciado de este problema. No es necesario que copies íntegramente el código `presentarTabla`; simplemente indica dónde colocarías la definición de la misma en el programa que escribas.

```
/*
 * Pre: ---
 * Post: Ha presentado en la pantalla la tabla de multiplicar del «n»:
 *
 *      LA TABLA DEL «n»
 *      «n» x 0 = 0
 *      «n» x 1 = «n»
 *      ...
 *      «n» x 9 = ...
 *      «n» x 10 = ...
 */
void presentarTabla(int n) {
    ...
}
```