



Escritura del calendario de un mes

Se desea disponer de un programa que escriba en la pantalla el calendario de un determinado mes de un año que son solicitados al usuario. El programa debe asegurarse de que el mes introducido está comprendido entre 1 y 12 y de que el año es igual o posterior a 1900, volviendo a solicitar los datos el número de veces que sea necesario hasta que cumplan con las restricciones establecidas. Los problemas que siguen a continuación van a guiar el diseño de este programa. Se facilita, además, el dato que el 1 de enero de 1900 fue lunes.

A continuación, se muestra un ejemplo de una posible ejecución del mismo, donde puede observarse tanto la petición y lectura de datos como la escritura del calendario elegido.

```
Introduzca el mes [1-12]: 13
El mes debe estar comprendido entre 1 y 12: -8
El mes debe estar comprendido entre 1 y 12: 0
El mes debe estar comprendido entre 1 y 12: 11
Introduzca un año igual o posterior a 1900: 1492
El año debe ser igual o posterior a 1900: 1899
El año debe ser igual o posterior a 1900: 2018
```

NOVIEMBRE 2019						
L	M	X	J	V	S	D

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Problema 1.º - Función main

Antes de comenzar a escribir la función main del programa solicitado, piensa en cómo podrías dividir el problema planteado en dos problemas relativamente independientes. Uno de ellos es bastante más grande que el otro y lo considerarás que mucho más difícil de resolver. Aun así, escribe la especificación y el cuerpo de la función main apoyándote en dos funciones que resuelvan cada uno de estos dos problemas. Escribe también la especificación y cabecera de esas dos funciones auxiliares. Su cuerpo será escrito posteriormente.

Apoyándote en esas dos funciones, el cuerpo de la función main puede llegar a constar simplemente de cuatro líneas de código.

Problema 2.º - Descomposición del problema que escribe el calendario

Posiblemente hayas llegado a la conclusión de que los dos problemas a los que se hace referencia anteriormente son, por un lado, la petición y lectura de los datos necesarios y que cumplan con las restricciones planteadas y, por otro, la escritura del calendario en la pantalla.

Con respecto a la función que va a resolver el problema de la escritura del calendario en la pantalla, **identifica los pasos que habría que seguir** para hacerlo. Evidentemente, algunos son más simples que otros. De entre ellos, **¿cuál o cuáles consideras que son los más complicados?** Considera que los pasos más complejos van a resolver problemas para los que será muy adecuado utilizar más funciones auxiliares. En este punto, basta con identificar y definir de forma precisa el subproblema que se va a resolver junto con sus entradas y salidas (esto es, especificando una función a través de su precondition, postcondition y cabecera).

Tu profesor o profesora va a guiar este proceso de identificación de problemas cada vez más simples en la pizarra. Este proceso se denomina **diseño descendente**. En esta clase de problemas vamos a aplicarlo de



Problemas de Programación 1 Diseño descendente

forma muy guiada. En los siguientes problemas se te va a pedir implementar funciones que resuelven problemas cada vez de menor entidad. **Echa un vistazo en primer lugar a las especificaciones y cabeceras de todas las funciones que se te solicita implementar** y cuando vayas a escribir el cuerpo de una de ellas, apóyate todo lo que puedas en la utilización de las funciones especificadas en los problemas siguientes, cuyos cuerpos escribirás con posterioridad.

Problema 3.º - Escritura del calendario

Diseña la función `escribirCalendario`, cuya especificación se muestra a continuación. **Apóyate sobre todo en las funciones `diaDeLaSemana`, `escribirCabecera` y `diasDelMes` especificadas en los problemas 4º, 5º y 7º.**

```
/*  
 * Pre: 1 ≤ mes ≤ 12 y agno ≥ 1900  
 * Post: Ha escrito en la pantalla el calendario del mes «mes» del año «agno» con el  
 *       formato mostrado al comienzo de este enunciado.  
 */  
void escribirCalendario(const int mes, const int agno);
```

Problema 4.º - Cálculo del día de la semana de una fecha

Diseña la función `diaDeLaSemana`, cuya especificación se muestra a continuación. **Apóyate sobre todo en las funciones `diasDelAño` y `diaEnElAño` especificadas en el problema 5º.**

```
/*  
 * Pre: Los valores de los parámetros «dia», «mes» y «agno» representan  
 *       conjuntamente una fecha válida del calendario gregoriano igual o posterior  
 *       al 1 de enero de 1900.  
 * Post: Ha devuelto un entero que codifica el día de la semana de la fecha  
 *       representada por los valores de los parámetros «dia», «mes» y «agno», de  
 *       acuerdo con la siguiente codificación: el 0 codifica el lunes, el 1  
 *       codifica martes y así sucesivamente hasta el 6, que codifica el domingo.  
 */  
int diaDeLaSemana(const int dia, const int mes, const int agno);
```

Problema 5.º - Funciones que devuelven número de días

Escribe el código de las funciones `diasDelAño` y `diaEnElAño` cuya especificación se muestra a continuación, y que formarán parte del módulo de biblioteca fechas que implementarás en la práctica 3ª.

```
/*  
 * Pre: agno > 1582  
 * Post: Ha devuelto el número de días que tiene el año «agno».  
 *       Por ejemplo: diasDelAño(2018) devuelve 365 y diasDelAño(2020), 366.  
 */  
int diasDelAño(int agno);  
  
/*  
 * Pre: 1 ≤ dia ≤ 31, 1 ≤ mes ≤ 12 y agno > 1582 y la fecha formada por dia/mes/agno  
 *       es una fecha válida del calendario.  
 * Post: Ha devuelto el número de día del año de la fecha formada por dia/mes/agno.  
 *       Por ejemplo: diaEnElAño(1, 1, 2019) devuelve 1; diaEnElAño(31, 12, 2018)  
 *       devuelve 365; diaEnElAño(1, 2, 2019) devuelve 32 y  
 *       diaEnElAño(31, 12, 2020) devuelve 366.  
 */  
int diaEnElAño(int dia, int mes, int agno);
```



Problemas de Programación 1 Diseño descendente

En el diseño de estas funciones, conviene que hagas uso de las funciones `esBisiesto` y `diasDelMes` con las que ya se trabajó en sesiones de problemas anteriores y que encontrarás ya implementadas en el módulo fechas de la práctica 3ª:

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

/*
 * Pre:  1 ≤ mes ≤ 12 y agno > 1582.
 * Post: Ha devuelto el número de días del mes correspondiente al parámetro «mes»
 *       del año correspondiente al parámetro «agno».
 *       Por ejemplo: diasDelMes(10, 2018) devuelve 31,
 *                   diasDelMes(2, 2018) devuelve 28 y
 *                   diasDelMes(2, 2020) devuelve 29.
 */
int diasDelMes(int mes, int agno);
```

Problema 6.º - Función que pide los datos al usuario y los filtra

Escribe el código correspondiente a la siguiente función:

```
/*
 * Pre:  ---
 * Post: Ha solicitado reiteradamente al usuario un mes de un año igual o
 *       posterior a 1900, leyéndolos en los parámetros «mes» y «agno», hasta que
 *       1 ≤ mes ≤ 12 y hasta que agno ≥ 1900.
 */
void pedirDatos(int& mes, int& agno);
```

Problema 7.º - Función que escribe la cabecera del calendario

Escribe el código de la función `escribirCabecera`:

```
/*
 * Pre:  1 ≤ mes ≤ 12
 * Post: Ha escrito en la pantalla la cabecera del calendario del mes «mes» del
 *       año «agno» con un el formato del siguiente ejemplo:
 *
 *           NOVIEMBRE 2019
 *       L   M   X   J   V   S   D
 *       -----
 */
void escribirCabecera(const int mes, const int agno);
```

En el diseño de la función `escribirCabecera`, puedes apoyarte en la función `escribirNombreMes`, cuya especificación aparece a continuación y cuya implementación se verá en el tema 10 del curso (cadenas de caracteres).

```
/*
 * Pre:  1 ≤ mes ≤ 12
 * Post: Ha escrito en la pantalla el nombre (en mayúsculas) del mes correspondiente
 *       al valor del parámetro «mes».
 */
void escribirNombreMes(const int mes);
```