



Prueba 1.^a voluntaria de evaluación

Esta prueba está dirigida a los alumnos matriculados en la asignatura de Programación 1 en el curso 2018-19. Consiste en completar el diseño e implementación del programa de escritura del calendario de un mes trabajado en la clase de problemas del día 29 de octubre de 2018. El resultado del trabajo se presentará antes del final del día 18 de noviembre a través de la plataforma Moodle, tal como se detalla al final de este documento.

Trabajo a realizar: escritura del calendario de un mes

En esta prueba voluntaria de evaluación se debe completar el diseño y la implementación en C++ del programa de escritura del calendario de un mes y año seleccionados por el usuario. Este problema ya se trabajó en la clase de problemas del día 29 de octubre de 2018 y su enunciado puede consultarse a través de este [enlace](#). Se debe continuar con el diseño descendente planteado en la clase y plasmado en el enunciado del problema.

Se facilita a continuación un borrador del fichero «calendario.cpp» que conforma el módulo principal del programa y que puede descargarse desde la sección «Código C++ y datos» de la página web de la asignatura, en el directorio «Prueba1».

```
/*
*****
* Curso de Programación 1. Tema 7 y prueba voluntaria de evaluación 1
* Autores: Miguel Ángel Latre, Javier Martínez y
*         AÑÁDETE COMO AUTOR (INDICANDO TAMBIÉN TU NIP)
* Última revisión: 29 de octubre de 2018
*         ACTUALIZA TAMBIÉN LA FECHA DE ÚLTIMA REVISIÓN
* Resumen: Programa que escribe en la pantalla el calendario de un determinado
*         mes de un año, que se solicitan al usuario. El programa se asegura de
*         que el mes introducido está comprendido entre 1 y 12 y de que el año
*         es igual o posterior a 1900.
*         Clase de problemas correspondiente al tema 7 (diseño modular y
*         descendente) y 1ª prueba voluntaria de evaluación (curso 2018-19).
*****
*/

#include <iostream>
#include <iomanip>
#include "../Biblioteca/fechas.h"
using namespace std;

/*
* 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) {
    /** Completar su código **/
}
```



```
/*
 * 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) {
    /** Completar su código **/
}

/*
 * Pre:  $1 \leq \text{mes} \leq 12$ 
 * Post: Ha escrito en la pantalla el nombre (en mayúsculas) del mes
 * correspondiente al valor del parámetro «mes».
 * Nota: Este código se explicará adecuadamente en el tema 10 (cadenas de
 * caracteres).
 */
void escribirNombreMes(const int mes) {
    const int NUM_MESES = 12;
    const int MAX_LONG_NOMBRE_MES = 10;
    const char NOMBRES_MES[NUM_MESES][MAX_LONG_NOMBRE_MES + 1] = { "ENERO",
        "FEBRERO", "MARZO", "ABRIL", "MAYO", "JUNIO", "JULIO", "AGOSTO",
        "SEPTIEMBRE", "OCTUBRE", "NOVIEMBRE", "DICIEMBRE" };
    cout << NOMBRES_MES[mes - 1];
}

/*
 * Pre:  $1 \leq \text{mes} \leq 12$ 
 * Post: Ha escrito en la pantalla la cabecera del calendario del mes «mes» del
 * año «agno» con un el formato del siguiente ejemplo:
 *
 *          NOVIEMBRE 2018
 *    L   M   X   J   V   S   D
 *    -----
 */
void escribirCabecera(const int mes, const int agno) {
    /** Completar su código **/
}

/*
 * Pre:  $1 \leq \text{mes} \leq 12$  y  $\text{agno} \geq 1900$ 
 * Post: Ha escrito en la pantalla el calendario del mes «mes» del año «agno» con
 * el formato del siguiente ejemplo:
 *
 *          NOVIEMBRE 2018
 *    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
 */
void escribirCalendario(const int mes, const int agno) {
    /** Completar su código **/
}
```



```
/*  
 * Pre: ---  
 * Post: Ha solicitado al usuario un mes de un año igual o posterior a 1900, se  
 *       ha asegurado de que el mes introducido está comprendido entre 1 y 12 y  
 *       de que el año es igual o posterior a 1900 y ha terminado escribiendo el  
 *       calendario del mes solicitado en la pantalla.  
 */  
int main() {  
    int mes, agno;  
    pedirDatos(mes, agno);  
    escribirCalendario(mes, agno);  
    return 0;  
}
```

Se puede observar que hay que implementar las funciones `pedirDatos`, `diaDeLaSemana`, `escribirCabecera` y `escribirCalendario`, mientras que las funciones `escribirNombreMes` y `main` ya están implementadas. También se deberá escribir el código de las funciones que se solicitan en el módulo de biblioteca `fechas` de la 3ª práctica, del que depende el módulo principal «`calendario.cpp`». Se recomienda leer el enunciado de dicha práctica antes de resolver esta prueba voluntaria de evaluación para localizar de qué funciones se trata.

El desarrollo del programa pedido debe hacerse en un proyecto denominado «Prueba1» ubicado en un directorio del mismo nombre. Este proyecto puede integrarse en el área de trabajo «Práctica 3» (de forma que «Prueba1» sea un subdirectorio del directorio «Practica3»). En cualquier caso, debe ponerse el cuidado necesario en la gestión de las áreas de trabajo, proyectos y directorios para que el programa compile sin necesidad de modificar la ruta relativa de inclusión del fichero de interfaz «`fechas.h`». Tampoco está permitido eliminar la directiva `#include` de dicho módulo y copiar y pegar código de funciones de dicho módulo en el módulo principal «`calendario.cpp`».

Tienes libertad para añadir más funciones auxiliares al módulo principal «`calendario.cpp`», si lo necesitas, **pero no** al módulo `fechas`. Igualmente, en el módulo principal, tienes libertad para modificar las cabeceras de las funciones resultantes de la descomposición modular, **pero no** la de la función `diaDeLaSemana`, que debe mantener su cabecera y especificación (incluyendo la codificación establecida para los días de la semana)¹.

Recuerda indicar tu nombre y NIP en el trabajo (añadiéndote como autor en la cabecera) y actualizar la fecha de última modificación.

Presentación del trabajo y valoración de la prueba

Presentación del trabajo

Cada alumno presentará dos ficheros con el código fuente C++ resultado de su trabajo en esta prueba a través de la plataforma Moodle (<https://moodle2.unizar.es/>) antes de las 24 horas del domingo 18 de noviembre de 2018. Los ficheros a presentar son:

- El fichero de implementación «`fechas.cpp`» del módulo `fechas` cuya implementación se solicita en la sección 3.3.5 de la práctica 3.
- El módulo principal del programa, de nombre «`calendario.cpp`», que contenga la función `main` y las funciones auxiliares en las que se apoye que no pertenezcan al módulo `fechas`.

Los trabajos no presentados a través de la plataforma Moodle o entregados fuera de plazo se considerarán no presentados.

¹ Estas restricciones sobre lo que es y no es susceptible de modificación son debidas a que parte de la corrección de la prueba se hará de forma muy exhaustiva y automática. No seguir las indicaciones dadas en este enunciado podría acarrear que la calificación de esta parte de la corrección sea 0.



Valoración de la prueba

El código presentado ha de estar libre de errores de compilación. El programa, al ser ejecutado, debe cumplir con los requisitos establecidos en el enunciado de la clase de problemas del 29 de octubre de 2018 proporcionando los resultados esperados al ser ejecutado. Adicionalmente, en el fichero correspondiente al módulo principal del programa, debe existir una función `diaDeLaSemana` con exactamente la misma cabecera que la presentada en este enunciado y que se comporte también exactamente igual a lo especificado en la misma. En caso contrario, tanto si el programa no se puede ejecutar o produce resultados muy alejados de lo pedido o si la función `diaDeLaSemana` no existe, la valoración del trabajo será la calificación mínima (0 sobre 10 puntos).

Los trabajos en los que el programa y la función `diaDeLaSemana` se comporten de acuerdo a sus especificaciones serán calificados sobre 10 puntos en función de su comportamiento (el programa será ejecutado manualmente y la función `diaDeLaSemana` será invocada de forma automática con un conjunto de casos de prueba muy exhaustivo), la calidad de su diseño (es decir, que sigan las pautas metodológicas explicadas en la asignatura) y la legibilidad de su código (conviene tener presentes las normas de la *Guía de estilo para programar en C++*).

Este trabajo es personal ya que sirve para evaluar el aprendizaje de cada alumno. No se admiten trabajos en equipo y, mucho menos aún, copias o plagios totales o parciales del trabajo.