

Programación y Estructuras de Datos (PED)

Examen sobre prácticas Junio 2021

Condiciones de entrega

- El examen se entrega a través del servidor de prácticas del DLSI <http://pracdlsi.dlsi.ua.es>. Tras cada entrega, el servidor enviará al alumno un **INFORME DE COMPILACIÓN**, para que el alumno compruebe que lo que ha entregado cumple las especificaciones pedidas y que se ha podido generar el ejecutable correctamente. **Este informe también se podrá consultar desde la página web de entrega de prácticas del DLSI** (<http://pracdlsi.dlsi.ua.es> e introducir el nombre de usuario y password). **SE ACONSEJA** que se haga una entrega previa con al menos el prototipo de la función pedida, para comprobar que no hayan errores de formato. Es posible que el servidor tarde más tiempo en devolver los informes en la última media hora del examen: **LO CUAL NO SERÁ ARGUMENTO** para justificar errores

- En caso que la práctica esté correctamente entregada, compilada y ejecutada, en este informe debe salir lo siguiente:

```
=====
DIFERENCIA CON FICHERO DE SALIDA DE REFERENCIA
=====
```

- En el servidor, el examen se comprobará con la siguiente función main:

```
TABBCalendario A; TListaCalendario L; TVectorCalendario V;
V = A.ABBCamino(L);
cout << V << endl;
```

- Esperando la salida:

```
[ ]
```

- Se tiene que entregar un fichero comprimido **tgz** (**tar cvzf fichero.tgz ***) que contenga todos los ficheros de los cuadernillos (con la estructura de directorios especificada en el enunciado de la práctica: **dentro del .tgz solo deben aparecer el fichero nombres.txt y los directorios lib, include y src**), junto con los métodos pedidos en el examen. El examen debe compilar con todos los ficheros entregados.
- El fichero **nombres.txt** tiene que contener el nombre del único autor del examen.
- El nombre de la función implementada por el alumno debe coincidir EXACTAMENTE con el prototipo propuesto en el enunciado.
- El alumno tiene que implementar su propio fichero de prueba (**tad.cpp**) para comprobar el código implementado (**este fichero no es necesario entregarlo**). Tampoco es necesario que se entregue el fichero makefile.
- El alumno **puede añadir a la parte privada las variables y métodos** que considere necesarios para la implementación.
- SI SE ENTREGA ALGO QUE NO COMPILA SUPONDRÁ UN CERO EN EL EXAMEN. Solo se evaluará la salida del programa. Se compilará con la versión del compilador instalada en los laboratorios de la EPS.**

- ARCHIVOS A ENTREGAR** (incluyendo en el código la función ABBCamino) :

```
include: tcalendario.h, tvectorcalendario.h, tlistacalendario.h, tabbcalendario.h
lib: tcalendario.cpp, tvectorcalendario.cpp, tlistacalendario.cpp, tabbcalendario.cpp
```

Añadir la siguiente función a la parte pública del **TABBCalendario**:

TVectorCalendario **ABBCamino** (**TListaCalendario** &)

El método **ABBCamino** invocado por un árbol ABB **TABBCalendario**, recibe una lista de **TCalendario** (**TListaCalendario**). Y devuelve un vector de **TCalendario** (**TVectorCalendario**).

El nuevo método tiene que efectuar lo siguiente:

- Recorrer la **TListaCalendario** de izquierda a derecha.
- Devolver un **TVectorCalendario** cuyas posiciones están todas ocupadas, es decir, no debe contener posiciones vacías. Para ello el **TVectorCalendario** deberá tener la dimensión correcta (por ejemplo, utilizando el método Redimensionar implementado en el cuadernillo de prácticas).
- Por cada elemento **TCalendario** de la lista, comprobar en el árbol **TABBCalendario** que invoca al método si el **TCalendario** existe en el árbol ABB, y operar así:
 - ✓ Si NO EXISTE en el árbol:
 - INSERTARLO en el árbol. Una vez insertado ...
 - ... INSERTAR al final del vector de salida **TVectorCalendario** los elementos **TCalendario** del árbol que pertenezcan al siguiente CAMINO:
 - desde la raíz del árbol hasta el **TCalendario** insertado, inclusive
 - ✓ Si EXISTE en el árbol:
 - No hacer nada

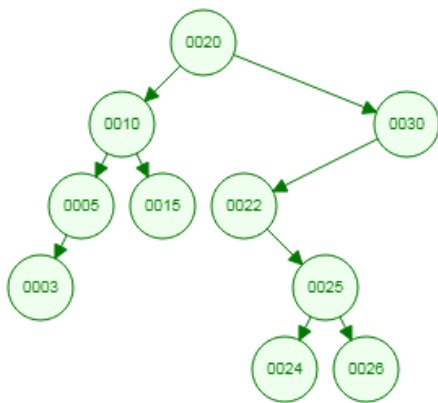
NOTAS:

- Si el elemento de la lista de entrada es un **TCalendario** vacío, no hacer nada sobre árbol y no hacer nada sobre la salida.
- Si la lista de entrada **TListaCalendario** es LISTA VACÍA, se devolverá un vector de salida **TVectorCalendario** vacío.
- Si el árbol **TABBCalendario** es ÁRBOL VACÍO, se devolverá un vector de salida **TVectorCalendario** vacío.

Ejemplo (se muestra sólo el DÍA DE LA FECHA de cada TCalendario, por simplificación):

LISTA ENTRADA: <21, 22, 31>

ABB inicial:



VECTOR SALIDA: [20, 30, 22, 21, 20, 30, 31]

ABB final:

