APELLIDOS: ***NOMBRE:***
TITULACIÓN: ***__GII __GII+ADE __GII+CRI __GII+MAT __GIS __GIS+GII __GIS+MAT***
DNI:
Duración: 1:45 h.

Se recuerda que, según la guía docente y para la convocatoria de mayo, el 15% de la nota final se corresponde con pruebas prácticas realizadas, 25% con prueba escrita teórica y 60% con prueba escrita práctica. Se sugiere dejar 1 hora de esfuerzo en la resolución del ejercicio 5. Contestar en la hoja de enunciados los 4 primeros ejercicios.

1) (**Prueba escrita teoría, 0.5 puntos**). Crear un árbol AVL dada la siguiente secuencia de números enteros: 5, 7, 9, 4, 6, 8, 14, 11, 12. Se deberán describir los pasos de cada inserción y cómo va quedando el árbol tras cada una de ellas.

2) (**Prueba escrita teoría, 0.5 punto**) De las diferentes implementaciones de listas dinámicas estudiadas en clase, enumerar cuáles de ellas se ajustan a la relación operación->orden de complejidad para todas las operaciones que se presentan a continuación. **Justifique su respuesta**.

   - Construir -> O(1)
   - InsertarFinal -> O(1)
   - Eliminar por cabecera -> O(1)
   - Pertenece -> O(n)

3) (**Prueba escrita teoría, 1.5 punto**) Considere el siguiente diagrama y estado de una implementación estática que simula el uso de memoria dinámica de un árbol binario de búsqueda para datos de tipo String:

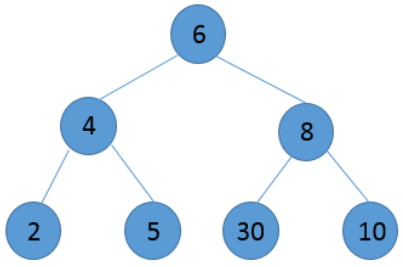
'Belen'		'David'		'Jesús'		'Juan'		'Soto'		'Alfonso'		'Almudena'	
7	2	0	0	1	5	0	6	0	0	2	0	0	0

cabLibres: 4
cabArbolBB: 3

a) Dibuje en forma arborescente cómo quedaría el árbol representado en memoria estática.

b) A continuación inserte el elemento de valor 'Antonio' mostrando cómo quedaría el contenedor en su representación estática.

4) (**Prueba escrita práctica, 1.5 puntos**) Dado un árbol binario cuyos nodos son de tipo `TElem` y éste se encuentra definido en la unidad `uElem` de la siguiente forma: `TElem=Integer.`



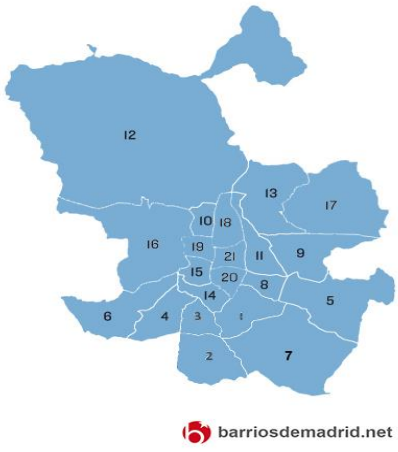
a) Se pide implementar una operación que devuelva el valor de la suma máxima de las ramas del árbol. En el árbol del ejemplo la suma máxima es 44.

b) Codificar toda la unidad `uRecorridos` que contenga los subprogramas necesarios para mostrar por pantalla los valores de todas las hojas de un árbol binario. Con el fin de evitar la violación de acceso y favorecer el encapsulamiento indique todas las operaciones y unidades necesarias para llevar a cabo este recorrido.

5) (**Prueba escrita práctica, 4.5 puntos**) Un empresario con diferentes negocios quiere embarcarse en uno nuevo: gimnasios *low cost*. Aunque este tipo de gimnasios lleva tiempo de moda, suelen utilizar instalaciones muy grandes, con muchas máquinas y salas disponibles para distintas actividades. Esto hace que no todos los barrios de una ciudad como Madrid tengan este tipo de gimnasios o los tengan cerca, por ello, el empresario ha decidido trasladar el concepto a todos los barrios que pueda, creando una red de gimnasios *low cost* de tamaño pequeño, los *small&low-cost gyms*. De esta forma, el tamaño de cada gimnasio vendrá dado por los locales disponibles que encuentre en cada barrio donde se vayan a instalar. No en todos los gimnasios se podrán realizar las mismas actividades (que dependerán del tamaño del gimnasio), aunque sí quiere asegurar que todos tengan un número razonable de máquinas para *fitness*. Los usuarios de la nueva marca de gimnasios, podrán acceder con la misma cuota de socios a cualquier gimnasio de la red, con independencia del barrio donde se encuentre, ya sea el distrito de su residencia, de su trabajo o cualquier otro.

Cada gimnasio estará caracterizado por un nombre, su dirección y el distrito en el que se ubica, y la lista de actividades que se pueden practicar en él (*Bodypump*, *Zumba*, *Burn*, *Yoga*, *Abs attack*, etc). Los diferentes gimnasios del mismo distrito están conectados entre sí, y también con los de los diferentes distritos contiguos. Por ejemplo, observando el mapa de los 21 distritos de Madrid, los gimnasios del distrito 1 estarán conectados con todos los gimnasios de los distritos 3, 5, 7, 8, 14 y 20.

Distritos	
1	Puente de Vallecas
2	Villaverde
3	Usera
4	Carabanchel
5	Vicálvaro
6	Latina
7	Villa de Vallecas
8	Moratalaz
9	San Blas-Canillejas
10	Tetuán
11	Ciudad Lineal
12	Fuencarral-El Pardo
13	Hortaleza
14	Arganzuela
15	Centro
16	Moncloa-Aravaca
17	Barajas
18	Chamartín
19	Chamberí
20	Retiro
21	Salamanca



Toda la información de los gimnasios abiertos por el empresario se encuentra almacenada en una base de datos. Para leer de la base de datos se proporciona para su uso (**y no hay que implementar**) el siguiente subprograma:

```
PROCEDURE ReadGymFromDB(VAR nombre: String;
      VAR direccion: TDireccion;
      VAR distrito: String;
      VAR actividades: TActividades;
      VAR error: Boolean);
```

Donde `TDireccion` está definido en la unidad `uDireccion` como:

```
TDireccion = RECORD
       calle: String;
       numero: Integer;
END
```

Y el tipo `TActividades` está definido en la unidad `uActividades` como:

```
TActividades = RECORD
       actividades: ARRAY[1..10] OF String;
       numeroActividades: Integer;
END;
```

El subprograma `ReadGymFromDB` devuelve en `nombre` el nombre del gimnasio, en `direccion` la información de la calle y número donde se encuentra el gimnasio que está leyendo en ese momento, en `distrito` el nombre del barrio donde se encuentra el gimnasio, en `actividades` las diferentes actividades que oferta el gimnasio y, por último, la devolución de la variable booleana `error` indicará cuándo se ha llegado al final de la consulta y no hay que seguir consultando información de gimnasios, por tanto devolverá `FALSE` indicando que se ha leído correctamente la información de un gimnasio (y por tanto no hay error) o `TRUE` cuando se haya llegado al final de la información y la lectura no contiene información útil. Se invocará al procedimiento `ReadGymFromDB` las veces que sean necesarias para ir cargando secuencialmente la información de los gimnasios desde la base de datos hasta que la variable error indique que no hay más información nueva que leer.

Se pide lo siguiente:

a) [1.5 punto] Definir el tipo de datos `TRedGimnasios` para representar la información en memoria principal de la red de gimnasios. Además de utilizar los tipos que se proporcionan, se deberán crear todos los tipos que se consideren necesarios para estructurar de forma adecuada la información. Cada tipo deberá estar en una unidad diferente.

   A continuación implementar un procedimiento (y su llamada en el programa principal) para cargar en memoria principal toda la información de la red de gimnasios almacenada en la base de datos. Si para ello se necesitan operaciones de los diferentes tipos de datos proporcionados, así como de los nuevos creados, deberán implementarse también en sus unidades correspondientes.

   Hay que tener en cuenta que para construir de forma adecuada la red, no sólo hay que utilizar el subprograma de consulta a la base de datos `ReadGymFromBD` para leer los datos de los diferente gimnasios, sino que también es necesario saber, una vez conocidos cuáles son todos los gimnasios de la red, qué distritos son contiguos y qué gimnasios tiene cada uno (para poder conectar unos gimnasios con otros). Esta información se puede obtener mediante el siguiente subprograma (se supone ya implementado):

```
PROCEDURE ReadContiguousDistrictsFromDB(nombre_gym: String;
          VAR distritosContiguos: TDistritosContiguos);
```

Donde la variable `distritosContiguos` devolverá el listado de distritos adyacentes al del distrito donde se encuentra el gimnasio de nombre pasado como entrada a través de la variable `nombre_gym`. El tipo `TDistritosContiguos` se encuentra definido en la unidad `uDistritosContiguos` como sigue:

```
TDistritosContiguos = RECORD
       distritos: ARRAY[1..21] OF String;
       numDistritos: Integer
END;
```

Y para consultar si un distrito está incluido en un listado `TDistritosContiguos` se dispone de la siguiente interfaz (**no hay que implementarlo**):

```
       FUNCTION ContieneDistrito(distritos: TDistritosContiguos;
                                 distrito: String): Boolean;
```

b) [1 punto] Implementar una operación que dado el nombre de una actividad, se pueda conocer el listado de gimnasios que ofrecen dicha actividad. Se deberán implementar las operaciones adicionales que se puedan necesitar.

c) [2 puntos] Implementar una operación que, dados los nombres de dos direcciones donde se encuentran dos gimnasios, devuelva el listado de distritos por los que habría que pasar para llegar de uno a otro.