

- Escribe tu **nombre y apellidos** en esta hoja e inmediatamente en todas las suplementarias, incluso las de sucio. El no hacerlo puede suponer tu expulsión
- Puedes utilizar el **lápiz** para tus respuestas. No está permitido el uso de apuntes, notas o libros. No puedes tener un **móvil** encendido, ni utilizar cualquier otro **aparato electrónico**.
- **Todos los alumnos implicados en una copia de un ejercicio tendrán una nota final de 0.** El alumno es responsable de velar por su examen. Es decir **tanto el que copia como el que se deja copiar (ya sea de manera activa o pasiva) recibirán el mismo castigo sin que exista atenuante alguno**

1. Ejercicio (3 puntos)

- a) ¿Cuál es la estructura de datos que utiliza un intérprete de un lenguaje de programación para llevar a cabo la recursión? ¿Por qué?

- b) Suponiendo que un árbol binario representa una expresión aritmética, **describe** que datos se almacenan en los nodos internos y externos. ¿Qué recorrido utilizarías para evaluar la expresión? ¿Por qué?

- c) ¿Cuál es la ventaja principal de la estructura de datos TablaHash respecto a un array, Vector o Lista enlazada? Y la desventaja principal?

- d) Una lista enlazada ¿Es una estructura de datos lineal o no-lineal?

- e) Problema práctico: La gerencia de un taller de autos quiere establecer un orden para el arreglo de los coches según la categoría de **fidelidad** del cliente y el orden de llegada del coche al taller. Para ello, categorizan a sus clientes de 1 a 5 según el gasto realizado (siendo el 1 el de mayor gasto, el más fiel). **Si en un momento se da el caso de que haya más de uno que tengan la misma categoría de fidelidad y ésta sea la mayor del momento, entonces el coche será elegido según el orden de llegada.** ¿Qué estructura de datos utilizarías para organizar los datos y poder elegir de forma eficiente el próximo coche para ser arreglado? ¿Por qué?

- f) Vamos a suponer que en un mismo programa se han definido dos instancias de la clase iteradora sobre una misma lista enlazada simple. ¿Se pueden producir problemas? En caso afirmativo, explica como o cuando puede ocurrir. En caso negativo, explica cual es la característica principal para que no ocurra ningún problema.

- Escribe tu **nombre y apellidos** en esta hoja e inmediatamente en todas las suplementarias, incluso las de sucio. El no hacerlo puede suponer tu expulsión
- Puedes utilizar el **lápiz** para tus respuestas. No está permitido el uso de apuntes, notas o libros. No puedes tener un **móvil** encendido, ni utilizar cualquier otro **aparato electrónico**.
- **Todos los alumnos implicados en una copia** de un ejercicio **tendrán una nota final de 0**. El alumno es responsable de velar por su examen. Es decir **tanto el que copia como el que se deja copiar (ya sea de manera activa o pasiva) recibirán el mismo castigo sin que exista atenuante alguno**

2. Ejercicio (2 puntos)

Se pide:

- Implementar el algoritmo genérico `insertionSort`.
- Implementar un método que dado un array de `Personas`, ordene el array por el nombre de la persona y en orden descendente, utilizando el algoritmo genérico `insertionSort`. La clase `Persona` contiene la información del nombre y la edad.

3. Ejercicio (2 punto)

Se quiere ampliar la funcionalidad de las listas doblemente enlazadas, vistas en clase, con el método `eliminarSig`. Dicho método eliminará el elemento situado tras la posición actual (`current`). ¿Cómo se tratan los posibles errores?

Se pide:

- Implementar las clases necesarias (sin los métodos) para representar una lista doblemente enlazada.
- Implementar el método `eliminarSig` y cualquier otro que se utilice.
- Describir gráficamente las siguientes situaciones, utilizando distintos ejemplos:
 - ¿Cómo se realiza la eliminación?
 - ¿Cómo se tratan los posibles errores?

4. Ejercicio (3 puntos)

Mediante un programa se puede generar un árbol binario de autoedición. Esto se podría hacer asignando una coordenada (x, y) a cada nodo del árbol. Dibujando un círculo alrededor de dicha coordenada y conectando cada nodo distinto de la raíz con su padre. Supongamos que disponemos de un árbol binario almacenado en memoria y que cada nodo contiene un objeto de tipo `Coordenada`, el cual contiene los atributos `x` e `y`. Supongamos que (0, 0) se corresponde con la esquina inferior izquierda. (ver Figura 1a)

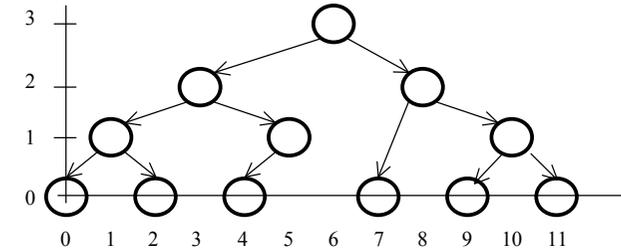
Se pide especificar, diseñar e implementar el método `calcularCoordenadas`, el cual calcula y etiqueta (ver Figura 1b) las coordenadas (x, y) de todos los nodos de la siguiente forma:

- La coordenada `x` se calcula usando el número que le corresponde en el recorrido en orden simétrico (`EnOrden`), de forma que para cada nodo `N`, todos los elementos del subárbol izquierdo de `N` aparecen a su izquierda y todos los elementos del subárbol derecho de `N` aparecen a su derecha.
- La coordenada `y` se calcula usando la altura del nodo.

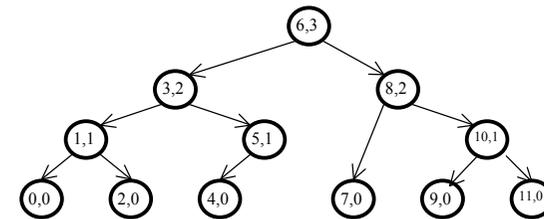
y además, muestra en pantalla cuales serán las dimensiones del árbol (anchura y altura) en una unidad imaginaria (ver Figura 1c).

Por ejemplo,

a) El árbol de entrada es el siguiente:



b) El árbol de salida es el siguiente:



c) La pantalla muestra las dimensiones (anchura x altura) del árbol:

Las dimensiones son: (11 u. x 3 u.)

Figura 1. a) Árbol binario que representa una figura con círculos pequeños. b) Árbol binario etiquetado con las coordenadas de todos los centros de dichos círculos. c) Resultado de las dimensiones del árbol.

Para la representación del árbol se provee de las siguientes clases:

<pre>public class BinTree{ protected BTNode root; public BTNode getRoot(); }</pre>	<pre>public class BinTreeItr{ protected BinTree bTree; }</pre>
<pre>public class BTNode{ protected Object content; protected BTNode left; protected BTNode right; // y los correspondientes get }</pre>	
<pre>public class Coordenada { private int x; private int y; // y los correspondientes get y set de // cada atributo }</pre>	<pre>public class CoordenadasBinTreeItr extends BinTreeItr{ ... }</pre>

Nota: Cualquier otro método que actúe sobre estas clases tendrá que ser implementado.