Ejercicio 1.- Extender la especificación de árbol general con las mismas operaciones del **Ejercicio 2** de **la hoja** árboles binarios, aunque aplicadas sobre árboles generales:

- igual_cantidad?: a_gen a_gen → bool, detecta si dos árboles generales tienen la misma cantidad de nodos;
- igual_forma?: a_gen a_gen → bool, que comprueba si dos árboles generales tienen la misma forma;
- suma_árboles: a_gen a_gen → bool, que recibe dos árboles generales de naturales y si tienen la misma forma genera el árbol resultante de sumar los valores de los nodos que ocupan el mismo lugar relativo en cada uno de ellos;
- cuenta_veces: a_gen a_gen → natural, que recibe dos árboles generales y devuelve la cantidad de veces que aparecen los nodos del primero en cualquier posición del segundo.

Ejercicio 2.- Llamaremos a un árbol general de naturales "maestro" si el valor de cada nodo es igual al número de hijos que tiene dicho nodo. Se pide:

a) Especificar completamente el TAD árbol general,
b) Comprobar si un árbol general es "maestro",
c) Buscar el nodo con mayor valor de un árbol maestro (es decir, el que tenga más hijos).

CRITERIOS PARA LA EVALUACIÓN DE LOS EJERCICIOS

EJERCICIO 1	**SI**	**NO**
1. Los algoritmos utilizan únicamente las operaciones del TAD ARBOL GENERAL. En el caso de usar otras operaciones auxiliares, estas se han definido previamente.		
2. Para contar el número de nodos de un AG, operación auxiliar necesaria, se ha implementado una primera operación para el AG y otra para el bosque o lista de hijos del AG: a. La primera llama a la segunda para contar el número de nodos del bosque. b. En la segunda se han tenido en cuenta todos los casos, bosque vacío y bosque no vacío, y se llama a la primera con el primer árbol del bosque, si existe, y a la segunda con el resto del bosque.		
3. Para comprobar si dos AG tienen la misma forma, se ha implementado una primera operación para comparar los AG`s y otra para comparar los bosques o listas de hijos de los AG's: a. La operación para AG llama a la segunda operación con los bosques o listas de hijos de ambos AG's. b. En la segunda operación se ha tenido en cuenta que ambos bosques pueden ser o no vacíos, y se llama a la operación para AG con los primeros árboles de ambos bosques y a la operación para bosques con el resto de ambos bosques.		
4. Para obtener el AG suma de dos AG de naturales, de igual forma, se ha implementado una primera operación para sumar dos AG's y otra para sumar dos bosques o listas de AG: a. La primera genera y devuelve un AG con la misma forma que los que recibe como entrada, con la raíz suma de las raíces de ambos y el bosque resultante de sumar los bosques o listas de ambos llamando a la segunda. b. En la segunda se han tenido en cuenta todos los casos, que sean bosques vacíos o bosques no vacíos, y se llama a la primera para sumar los primeros AG de cada uno de los bosques, si existen, y a la segunda para sumar el resto de ambos bosques.		
5. Para comprobar si un elemento esta en uno de los nodos de un AG, operación auxiliar necesaria para contar el número de veces que aparecen nodos de un AG en otro, se ha implementado una operación para ver si un elemento está en un AG y otra para ver si un elemento está en un bosque o lista de AG's: a. La primera comprueba si el elemento está en la raíz del árbol y, si es necesario, llama a la operación para bosque. b. En la segunda se han tenido en cuenta todos los casos, bosque vacío y bosque no vacío, y se llama a la primera con el primer árbol del bosque y, si es necesario, a la segunda con el resto del bosque.		
6. Para contar el número de veces que aparecen nodos de un AG en otro se implementan dos operaciones, una para AG y otra para el bosque o lista de hijos: a. La primera comprueba si el elemento de la raíz del AG está en el segundo AG, llamando a la operación auxiliar y después llama a la función que cuenta en el bosque. b. La segunda, que cuenta los nodos del bosque que están en el segundo AG, comprueba si el bosque está vacío y, sino lo está llama a la primera con el primer árbol del bosque y a la segunda con el resto del mismo.		

EJERCICIO 2

1. Se ha descrito el TAD ARBOL GENERAL, las operaciones básicas y las ecuaciones que las describen, tal y como se detallan en las transparencias de clase.
2. Los algoritmos de los apartados b y c usan únicamente las operaciones del TAD ARBOL GENERAL. En el caso de usar otras operaciones auxiliares, estas se han definido previamente.
3. Para comprobar si un AG es maestro, se implementa una operación para el AG y otra para el bosque:
    a. La primera comprueba si el valor del nodo raíz es la longitud del bosque y, si lo es, llama a la segunda para comprobar el bosque.
    b. La segunda comprueba si el bosque es vacío y, si no lo es, llama a la primera con el primero de los árboles del bosque y a la del bosque con el resto del mismo.
4. Para calcular el valor máximo de un bosque maestro, se implementan dos funciones:
    a. La primera compara el valor del nodo raíz con el máximo del bosque obtenido llamando a la segunda.
    b. La segunda, máximo del bosque, tiene en cuenta que el bosque puede ser vacío y, si no lo es, calcula el máximo del primer árbol llamando a la primera y lo compara con el máximo del resto el bosque llamando a la segunda.