Ejercicios con estructuras enlazadas arborescentes

En todos los ejercicios de este listado se usarán las siguientes definiciones para representar árboles binarios mediante estructuras enlazadas. El árbol binario vacío se representará mediante el valor `NULL`:

```
typedef int Elemento;

struct Nodo {
    Elemento elem;
    struct Nodo * hijoIzquierdo;
    struct Nodo * hijoDerecho;
};

typedef struct Nodo * ArbolBinario;
```

1. Completa el código de la siguiente función para que construya y devuelva un árbol binario con un único nodo cuyo valor sea `d`.

   ```
   ArbolBinario crea( Elemento d )
   ```

2. Completa el código de la siguiente función para que construya y devuelva un árbol binario cuya raíz tenga el valor 1, cuyo hijo izquierdo el valor 2 y cuyo hizo derecho el valor 3.

   ```
   ArbolBinario crea()
   ```

3. Completa el código de la siguiente función para que muestre todos los valores delárbol `a` recorriéndolo en preorden.

   ```
   void muestra( ArbolBinario a )
   ```

4. Completa el código de la siguiente función para que muestre todos los valores del árbol `a` recorriéndolo en inorden.

   ```
   void muestra( ArbolBinario a )
   ```

5. Completa el código de la siguiente función para que muestre todos los valores del árbol `a` recorriéndolo en postorden.

   ```
   void muestra( ArbolBinario a )
   ```

6. Completa el código de la siguiente función para que libere toda la memoria reservada para el árbol `a`.

   ```
   void libera( ArbolBinario a )
   ```

7. Completa el código de la siguiente función para que devuelva el número de nodos del árbol binario `a`.

```
int cuenta( ArbolBinario a )
```

8. Completa el código de la siguiente función para que devuelva el número de hojas del árbol binario `a`.

```
int cuenta( ArbolBinario a )
```

9. Completa el código de la siguiente función para que devuelva el número de nodos internos del árbol binario `a`.

```
int cuenta( ArbolBinario a )
```

10. Completa el código de la siguiente función para que devuelva el número de nodos del árbol binario `a` cuyo valor sea mayor a `d`.

```
int cuenta( ArbolBinario a, Elemento d )
```

11. Completa el código de la siguiente función para que devuelva el número de hojas del árbol binario `a` cuyo valor esté en el intervalo [`m`, `n`].

```
int cuenta( ArbolBinario a, Elemento m, Elemento n )
```

12. Completa el código de la siguiente función para que devuelva el número de nodos internos del árbol binario `a` cuyo valor sea distinto a cualquiera de los `n` valores del array `m`.

```
int cuenta( ArbolBinario a, Elemento m[], int n )
```

13. Completa el código de la siguiente función para que devuelva la suma de los valores de todos los nodos del árbol binario `a`.

```
int cuenta( ArbolBinario a )
```

14. Completa el código de la siguiente función para que devuelva 1 si el árbol `a` contiene el elemento `d` y 0 en caso contrario.

```
int existe( ArbolBinario a, Elemento d )
```

15. Completa el código de la siguiente función para que devuelva el mayor de los valores guardados en los nodos del árbol binario `a`, suponiendo que todos los valores guardados son mayores que cero.

```
int maximo( ArbolBinario a )
```

16. Completa el código de la siguiente función para que devuelva la altura del árbol binario `a`, es decir, la longitud del camino más largo de la raíz a cualquiera de sus descendientes.

```
int altura( ArbolBinario a )
```

17. Completa el código de la siguiente función para que añada a cada hoja dos nuevos hijos, el izquierdo tendrá como valor el doble de su padre y el derecho el doble más uno.

```
void amplia( ArbolBinario a )
```

18. Completa el código de la siguiente función para que devuelva 1, si los dos árboles binarios pasados como parámetro son iguales y 0 en caso contrario.

```
int compara( ArbolBinario a, ArbolBinario b )
```

19. Completa el código de la siguiente función para que construya y devuelva un árbol binario idéntico al pasado como parámetro.

```
ArbolBinario copia( ArbolBinario a )
```

20. Usando las siguientes definiciones para implementar una estructura enlazada lineal

```
struct Celda {
      Elemento elem;
      struct Celda * sig;
};

typedef struct Celda * CeldaPtr;
```

Completa el código de la siguiente función para que devuelva una lista con todos los elementos contenidos en el árbol binario `a`.

```
CeldaPtr enumera( ArbolBinario a )
```

21. Repite todos los ejercicios anteriores (excepto el 4) usando las siguientes definiciones para representar árboles generales mediante estructuras enlazadas con la representación primer hijo – hermano derecho.

```
typedef int Elemento;

struct Nodo {
      Elemento elem;
      struct Nodo * primerHijo;
      struct Nodo * hermanoDerecho;
};

typedef struct Nodo * ArbolGeneral;
```