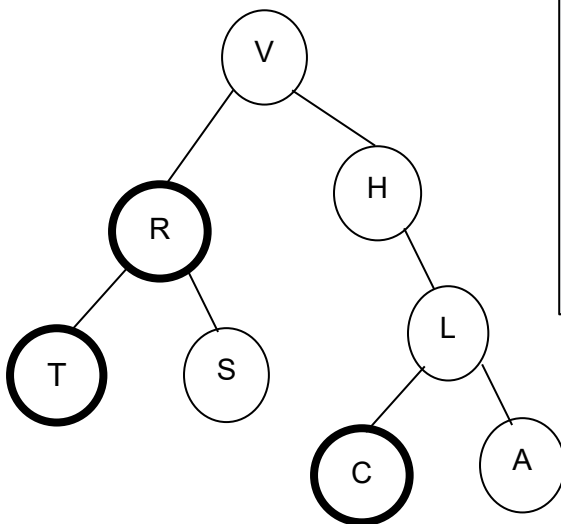


## Nodos Izquierdos

Implementa una función en el `TADArbolBinario` para que, dado un árbol de este tipo, te devuelva una lista (no importa el orden) con los elementos del árbol que son hijos izquierdos de un nodo cualquiera del mismo. Como pequeña pista para la realización del ejercicio recordad que, aunque la función que aparece en el `.h` tiene que tener una determinada interfaz (la que aparece en el proyecto facilitado para realizar el examen) luego, en el `.c`, se pueden definir funciones auxiliares privadas (ya no aparecen en el `.h`) que ayuden a implementar esta función inicial. En el siguiente ejemplo se puede ver un árbol binario y la ejecución del método sobre dicho árbol, donde los nodos izquierdos son aquellos resaltados con un trazo más grueso. Este árbol de la imagen es el que se proporciona en el proyecto del examen, para que el alumno pueda probar su implementación con él. No obstante, se recomienda modificarlo para poder probar su función con otros ejemplos.



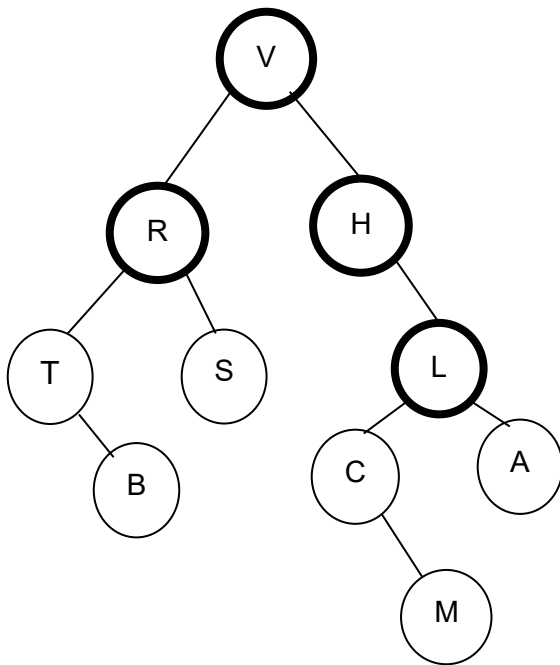
Son hijos izquierdos

CTR

Process finished with exit code 0

## Nodos Abuelos

Implementa una función en el `TADArbolBinario` para que, dado un árbol de este tipo, te devuelva una lista con los elementos del árbol que son abuelos junto al número de nietos. En este caso el orden de los elementos a mostrar tiene que ser por niveles y, dentro de estos, de izquierda a derecha. En el siguiente ejemplo se puede ver un árbol binario y la ejecución del método sobre dicho árbol, donde los nodos abuelos son aquellos resaltados con un trazo más grueso. Este árbol de la imagen es el que se proporciona en el proyecto del examen, para que el alumno pueda probar su implementación con él. No obstante, se recomienda modificarlo para poder probar su función con otros ejemplos. Como pequeña ayuda para la realización del ejercicio se incluye un segundo tipo de lista y de elemento en la carpeta del proyecto. Modifica este segundo tipo de elemento si lo ves necesario para resolver el problema.



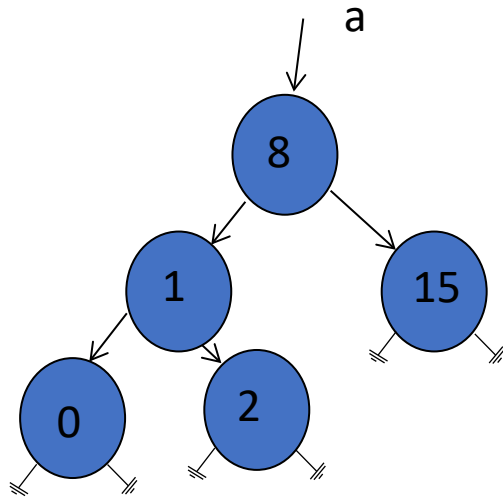
Son nodos abuelos

Lista: V (3) R (1) H (2) L (1)

Process finished with exit code 0

Implementa en el `TADArbol` que se te proporciona junto a este enunciado la función `ascendientes`, que recibe un puntero a un árbol y a un elemento (que puede o no estar en el árbol) y nos devuelve una lista con los nodos que hay desde la raíz hasta el nodo en cuestión. En caso de que el nodo no se encuentre en el árbol, devolverá una lista vacía.

```
void ascendientes (TBinarySearchTree a, TElemento e, TLista* r);
```



`ascendientes (a,2,lista) = [8,1,2]`

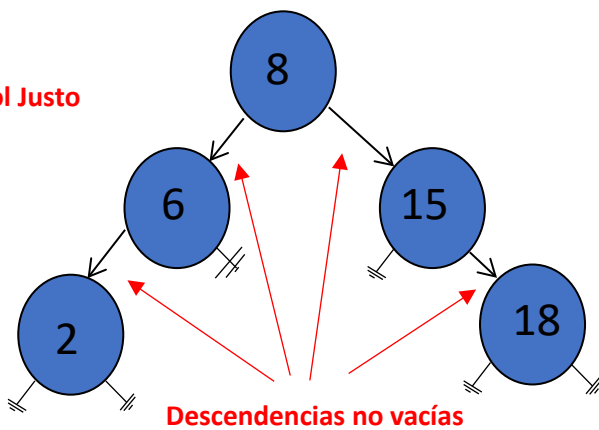
`ascendientes (a,9,lista) = []`

Implementa en el `TADArbolBinarioBusqueda` que se te proporciona junto a este enunciado la función:

- `arbolJusto`: que recibe un árbol `a` y nos dice si el árbol es justo. Entendemos por árbol justo aquel que, en su totalidad, tiene el mismo número de descendencias no vacías a su izquierda y a su derecha.

```
int arbolJusto(TBinarySearchTree a);
```

Árbol Justo



Árbol No Justo

