

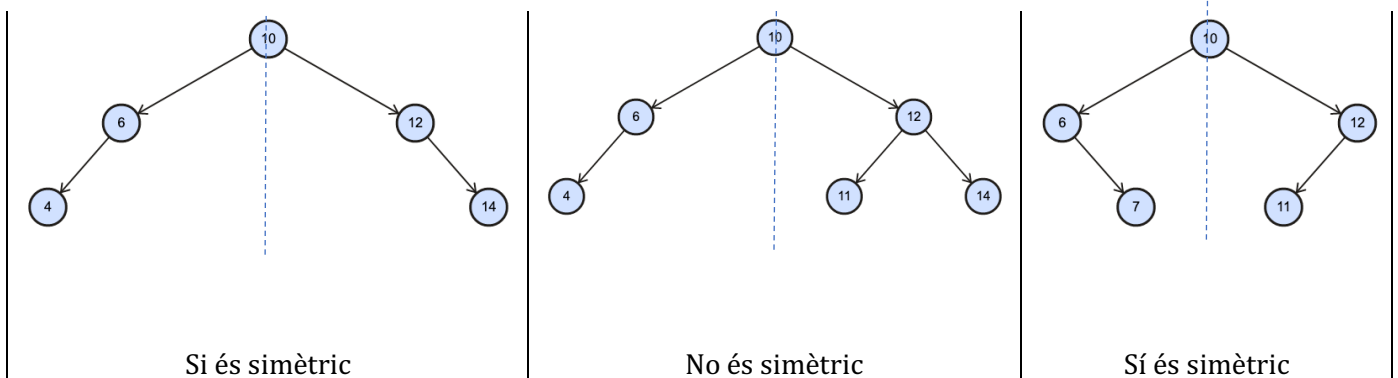
Nom i Cognoms: _____ Signatura: _____

Abans de començar:

- **CAMPUS VIRTUAL:**
 - Descarrega't el projecte que vas lliurar el diumenge dia 16/5/2021.
 - Descarrega't el main del campus virtual.
- En aquesta prova s'ha d'utilitzar **Netbeans 8.2, C++ versió 11**
- S'ha de lliurar al Campus Virtual **un únic fitxer zip** amb el vostre nom i cognoms. Per exemple **LisaSimpson_control_P3.zip**. **Aquest fitxer contindrà el codi de tots els exercicis del lliurament, incloent les funcions dels dos exercicis de la prova.**
- **Utilitzeu el projecte de l'exercici 1 com a projecte base per fer la prova**
 - Cada exercici de la prova és una nova funció a implementar en el TAD BinarySearchTree de l'exercici 1 de la pràctica.
 - Penseu que els exercicis estan explicats amb templates, en cas que hagueu implementat el TAD sense templates, elimineu els templates al vostre codi. En cas que tingueu la implementació amb templates, mantingueu la definició que se us indica.
 - En el vostre main.cpp, incorporeu les dues funcions per testejar cada exercici de la prova.

(3 p.) EXERCICI 1. És arbre simètric.

Donat un arbre binari de cerca, es demana implementar un mètode que determini si l'arbre és simètric a l'eix del root. És a dir, el subarbre de l'esquerra és un mirall del subarbre de la dreta. En el cas que l'arbre estigui buit o només tingui un node, sí és simètric. Mireu uns exemples:



Es demana el següent:

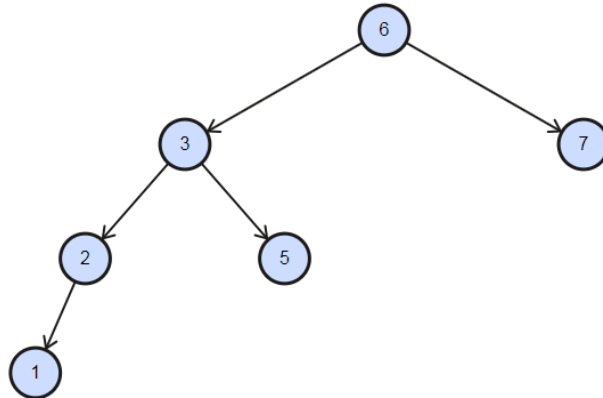
- Agafeu la classe BinarySearchTree i implementeu el mètode:


```
// El mètode retorna true en cas de que els subarbre esquerra sigui mirall del subarbre dret o fals en cas contrari.
// Aquesta funció ha de ser recursiva
template <class K, class V>
bool BinarySearchTree<K,V>::esArbreSimetric(){
    // codi a implementar
}
Definiu 1 funció auxiliar si us fa falta
```

No podeu modificar el main, només eliminar els templates si no en teniu.

(3 p.) EXERCICI 2. La suma màxima dels camins possibles a la fulla des de l'arrel.

Donat un arbre de cerca binari T es demana la suma màxima dels camins possibles des de l'arrel a les fulles. En el cas que l'arbre estigui buit, la suma màxima de camins és 0.



A l'exemple: els camins possibles són:

$$6+3+2+1 = 12$$

$$\mathbf{6+3+5 = 14}$$

$$6+7 = 13$$

Per tant, la suma màxima de tots els camins possibles donarà com a resultat 14.

Es demana el següent:

- Agafeu la classe BST i implementeu el mètode següent:

```
// Retorna la suma màxima d'arrel a fulla en un arbre binari de cerca
// Aquesta funció ha de ser recursiva
template <class K, class V>
int BinarySearchTree<K,V>::sumaMaximaAFulles()
{
    // codi a implementar
}
Definiu 1 funció auxiliar si us fa falta
```

No podeu modificar el main, només eliminar els templates si no en teniu.