

Nom i Cognoms: _____ Signatura: _____

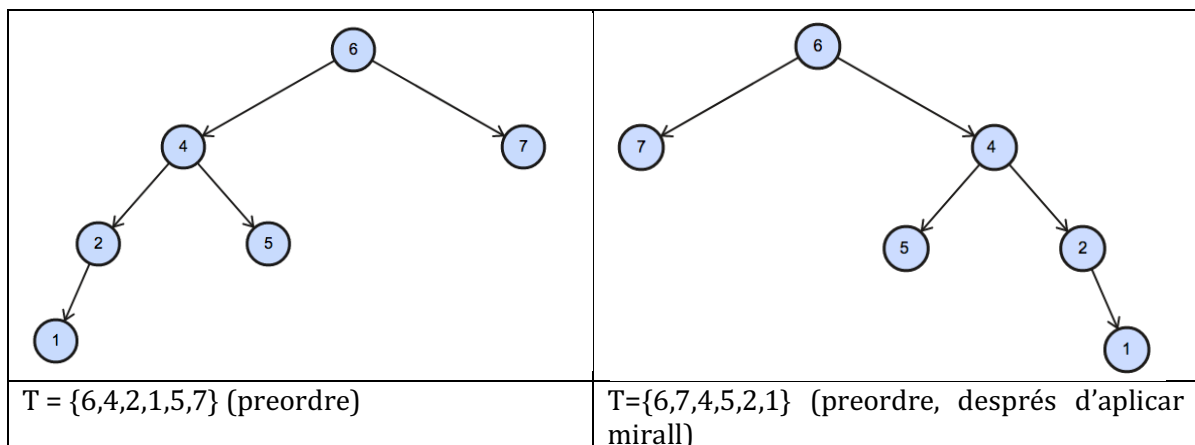
Abans de començar:

- **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.

(3.0 p.) EXERCICI 1. Arbre Mirall (*la traducción de mirall es espejo en castellano*).

Donat un arbre de cerca binaria, implementeu una funció recursiva que realitzi l'operació mirall sobre un arbre.

Al següent exemple s'observa aquesta operació. A l'arbre de l'esquerra, quan es crida a l'operació mirall queda com es visualitza l'arbre de la dreta. **Mireu amb atenció la funció que heu d'implementar i noteu que l'arbre no es retorna. No s'ha de crear un arbre auxiliar ni utilitzar cap altre estructura de dades per a la solució, sinó que s'ha de modificar directament l'arbre intern del TAD de forma recursiva.**



Es demana el següent:

- Agafeu la classe BinarySearchTree i implementeu els mètodes següents:

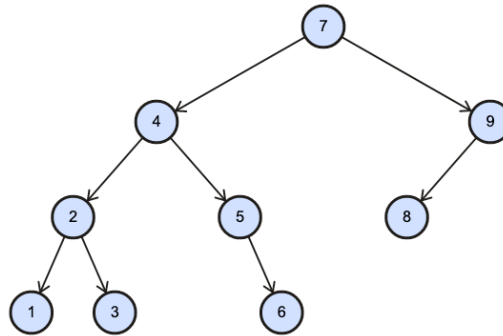
```
// El mètode aplica l'operació mirall sobre el arbre
template <class K, class V>
void BinarySearchTree<K,V>::mirall()
{
    // codi a implementar
}
```

Definiu 1 funció auxiliar si us fa falta

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

(3.0 p.) EXERCICI 2. Mostra les claus (keys) dels nodes que estan a nivell k

Donat un arbre de cerca binària T es demana mostrar tots els valors dels nodes que estiguin en el nivell K de l'arbre. Assumirem que k és un enter positiu.



A l'exemple:

- Si $k = 0$, el mètode mostrarà:
 - nivell 0 key 7
- Si $k = 1$, el mètode mostrarà
 - nivell 1 key 4
 - nivell 1 key 9
- Si $k = 2$, el mètode mostrarà
 - nivell 2 key 2
 - nivell 2 key 5
 - nivell 2 key 8
- Si $k = 3$, el mètode mostrarà
 - nivell 3 key 1
 - nivell 3 key 3
 - nivell 3 key 6
- Si $k = 4$, el mètode mostrarà
 - El mètode no mostra res per pantalla

Es demana el següent:

- Agafeu la classe BinarySearchTree.h i implementeu:

```
// Aquest mètode mostra les claus del nivell k d'un arbre
template <class K, class V>
void BinarySearchTree<K,V>::mostraNodesNivell(int k){
    // codi a implementar
}

// Aquesta funció us ha de servir per fer la recursivitat
template <class K, class V>
void BinarySearchTree<K,V>::mostraNodesNivell(BinaryTreeNode<K,V>* node, int k, int L )
{
    // codi a implementar
}
```

No podeu afegir més mètodes auxiliars i no podeu modificar el main o les funcions. Podeu eliminar els templates si no en teniu als vostres TADs.