

---

## Mètode de la classe `BinaryTree` que verifica si dos arbres binaris són idèntics

---

X69546\_ca

Implementa un nou mètode de la classe `BinaryTree` que verifica si dos arbres binaris són idèntics.

D'entre els fitxers que s'adjunten en aquest exercici, trobaràs `BinaryTree.old.hpp`, a on hi ha una implementació de la classe genèrica `BinaryTree`. En primer lloc, hauràs de fer:

```
cp BinaryTree.old.hpp BinaryTree.hpp
```

A continuació si obres el fitxer `BinaryTree.hpp` al final del mateix trobaràs el mètode que has d'implementar:

```
// Pre: cert
// Post: Torna true si el pi i t són idèntics i false en cas contrari.
bool operator==(const BinaryTree &t) const;
```

**IMPORTANT:** No toquis la resta de la implementació de la classe, excepte si necessites afegir algun mètode auxiliar o atribut a la part privada.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `program.cpp` (programa principal) i `Makefile` per a compilar i generar l'executable. El programa principal que t'oferim ja s'encarrega de llegir els arbres binaris i fer les crides al mètode indicat. **Només cal que implementis el mètode `operator==`.**

Per a pujar la teva solució, has de crear el fitxer `solution.tar` així:

```
tar cf solution.tar BinaryTree.hpp
```

### Observació 1

Hauries d'aconseguir implementar el mètode demanat usant només la part privada de la classe, és a dir, usant únicament la representació de la classe. L'ús dels mètodes públics de la classe serà fortament penalitzat.

### Observació 2

Recorda que si crees funcions auxiliars, has d'afegir-hi les corresponents **Precondició** (Pre) i **Postcondició** (Post). En els bucles inclou l'**invariant del bucle** (Inv) i la **funció de fita** (FF). En les crides recursives inclou la **hipòtesi d'inducció** (HI) i la **funció de fita** (FF).

### Entrada

Una seqüència de parells d'arbres binaris.

## Sortida

Per a cada parell d'arbres binaris s'escriurà el resultat del mètode `operator==`.

El programa principal que t'oferim ja s'encarrega de llegir la seqüència de parells d'arbres binaris i fer les crides corresponents al mètode de `BinaryTree` que se't demana d'implementar.

Només cal que facis les modificacions abans esmentades dins el fitxer `BinaryTree.hpp`.

Per més detalls de com és l'entrada i la sortida consulta els jocs de proves públics.

### Exemple d'entrada 1

```
1 == 1
2 == 1
() == 0
7(2,5) == 7(2,5)
5(,1) == 5(1,)
5(,1) == 5(,1(1,))
5(,1(1,)) == 5(,1(1,))
5(,1(,1)) == 5(,1(1,))
4(2(,3),2) == 5(1,)
1(3,4(3,2)) == 1(3,4(3,2))
3(1,4(3,2)) == 1(3,4(3,2))
2(5(2,2),) == 2(5(2,2),)
2(5(2,2(5,)),) == 2(5(2(2(5,)),),),)
12(5(2(2(5,)),,1),0) == 12(5(2(2(15,)),,1),0)
```

### Exemple de sortida 1

```
1 == 1 : true
2 == 1 : false
() == 0 : false
7(2,5) == 7(2,5) : true
5(,1) == 5(1,) : false
5(,1) == 5(,1(1,)) : false
5(,1(1,)) == 5(,1(1,)) : true
5(,1(,1)) == 5(,1(1,)) : false
4(2(,3),2) == 5(1,) : false
1(3,4(3,2)) == 1(3,4(3,2)) : true
3(1,4(3,2)) == 1(3,4(3,2)) : false
2(5(2,2),) == 2(5(2,2),) : true
2(5(2,2(5,)),) == 2(5(2(2(5,)),),),) : false
12(5(2(2(5,)),,1),0) == 12(5(2(2(15,)),,1),0) : false
```

## Informació del problema

Autoria: Bernardino Casas

Generació: 2026-01-25T21:22:34.616Z

© Jutge.org, 2006–2026.

<https://jutge.org>