Poda un arbre binari sense repeticions — U38261_ca

Implementeu la funció següent:

```
/**
 * @brief Poda les branques amb valor `x` d'un arbre binari
 *        sense repeticions de valors.
 *
 * Retorna un nou arbre binari que és una còpia de `t`
 * però sense les branques que contenen el valor `x`.
 * Alhora retorna si la poda ha tingut èxit o no s'ha podat res.
 *
 * @param t Arbre binari amb tots els valors diferents.
 * @param x Valor de les branques a podar.
 *
 * @pre `t` és un arbre binari a on tots els nodes tenen _valors diferents_.
 *
 * @return Retorna una parella (`std::pair`) amb l'arbre podat
 *         i un booleà que és `true` si s'ha podat alguna branca
 *         i `false` si l'arbre resultat és igual que `t`.
 */
std::pair<pro2::BinTree<int>, bool> prune_tree(pro2::BinTree<int> t, int x);
```

Observació

Els fitxers públics (icona del gatet) contenen:

`bintree.hh`	la classe `BinTree`
`bintree-io.hh`	l'entrada/sortida de `BinTree`
`prune.hh`	declaració de `prune_tree`
`main.cc`	el programa principal

També hi ha un `Makefile` i el directori `.vscode` que té la configuració per compilar i depurar amb VSCode.

Cal implementar `evaluate` en un **fitxer `.cc` nou**, compilar, i finalment **enviar només el fitxer amb la funció**.

Entrada

Cada cas consisteix en una representació textual d'un arbre d'enters (`t`) seguit d'un enter, que és el valor a podar (`x`). (Això ja ho fa el `main` que es dóna.)

Sortida

Per a cada cas, la sortida és l'arbre resultant de la poda, cada resultat en una línia separada. Si no s'ha podat res, s'escriu un missatge "No s'ha trobat `x`", amb el valor concret d'`x`. (Això també ho fa el `main` que es dóna.)

Exemple d'entrada 1

```
visual
3

1
|-- 2
'-- 3
    |-- 4
    '-- 5

5

1
|-- 2
|   |-- 3
|   '-- 4
'-- 5
    |-- #
    '-- 6

3

1
|-- 2
|   |-- 3
|   '-- 4
'-- 5
    |-- 6
    '-- 7
```

Exemple de sortida 1

```
1
|-- 2
'-- #


1
|-- 2
|   |-- 3
|   '-- 4
'-- #


1
|-- 2
|   |-- #
|   '-- 4
'-- 5
    |-- 6
    '-- 7
```

Exemple d'entrada 2

```
visual
1

1

4

1
|-- #
'-- 2
    |-- 3
    '-- 4

5

1
|-- 2
|   |-- 3
|   '-- 4
'-- 5
    |-- 6
    '-- 7

2

1
|-- #
```

```
'-- 2
    |-- 3
    '-- 4
```

Exemple de sortida 2

```
#


1
|-- #
'-- 2
    |-- 3
    '-- #
```

```
1
|-- 2
|   |-- 3
|   '-- 4
'-- #


1
```

Informació del problema

Autor : Borja Valles i Pau Fernández
Generació : 2025-03-21 20:35:33