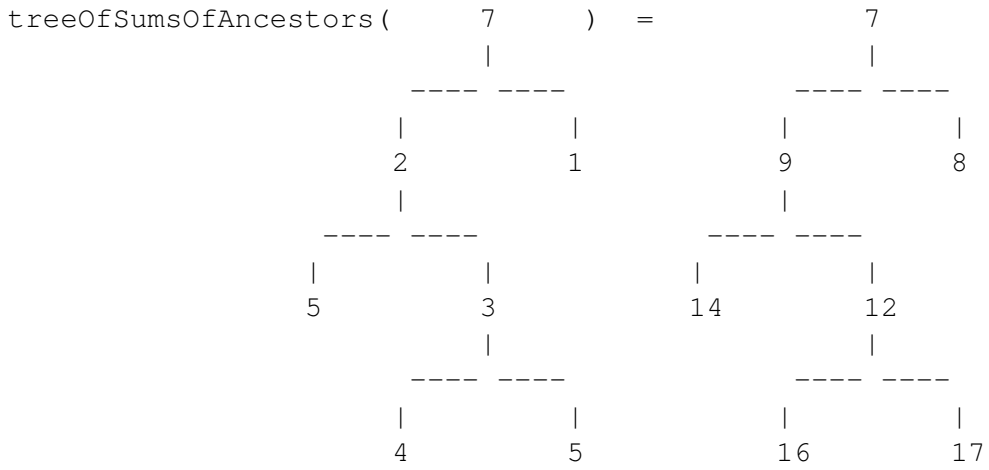
Arbre de sumes d'ancestres X53945_ca

Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna un nou arbre amb la mateixa estructura, i a on cada posició conté la suma del valor del propi node més els valors dels nodes dels ancestres d'aquella mateixa posició a l'arbre inicial. Aquesta és la capcelera:

```
// Pre:
// Post: Retorna un arbre d'enters t' amb la mateixa estructura que t.
//       Per a cada posició p, el valor guardat a t' a posició p és igual a la
//       dels valors guardats a t a posició p i a posicions ancestres de p.
BinTree<int> treeOfSumsOfAncestors(const BinTree<int> t);
```

Aquí tenim un exemple de paràmetre d'entrada de la funció i la corresponent sortida:



Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc`, `BinTree.hh`, `treeOfSumsOfAncestors.hh`. Us falta crear el fitxer `treeOfSumsOfAncestors.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Només cal que pugeu `treeOfSumsOfAncestors.cc` al jutge.

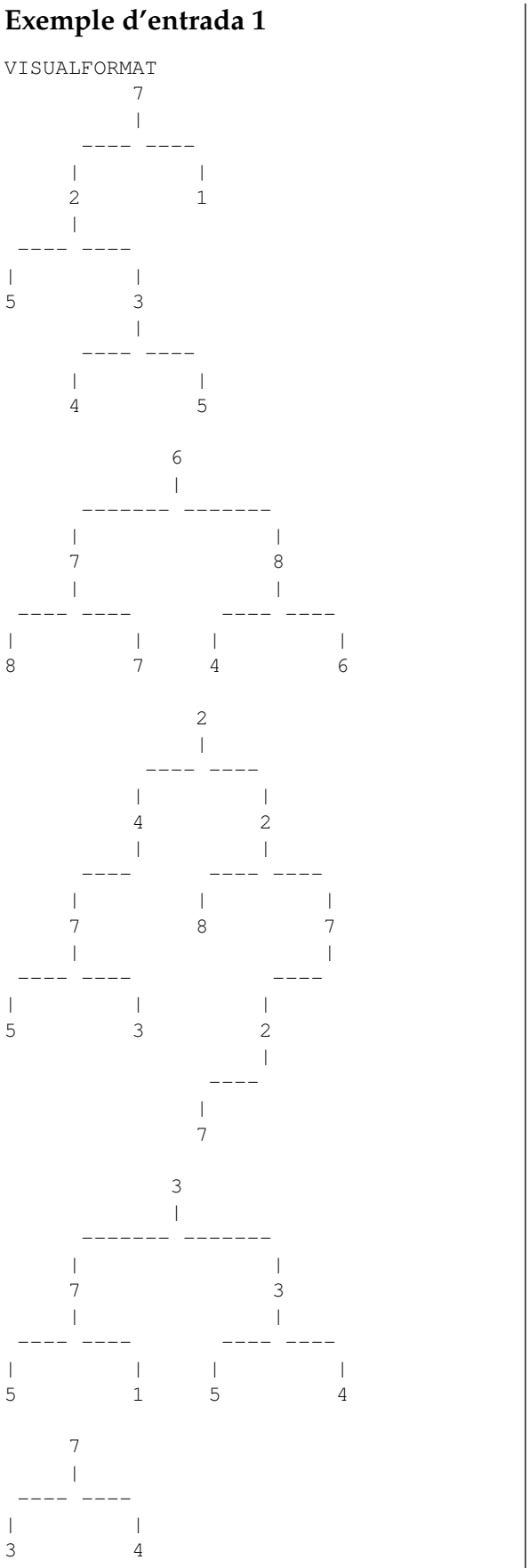
Entrada

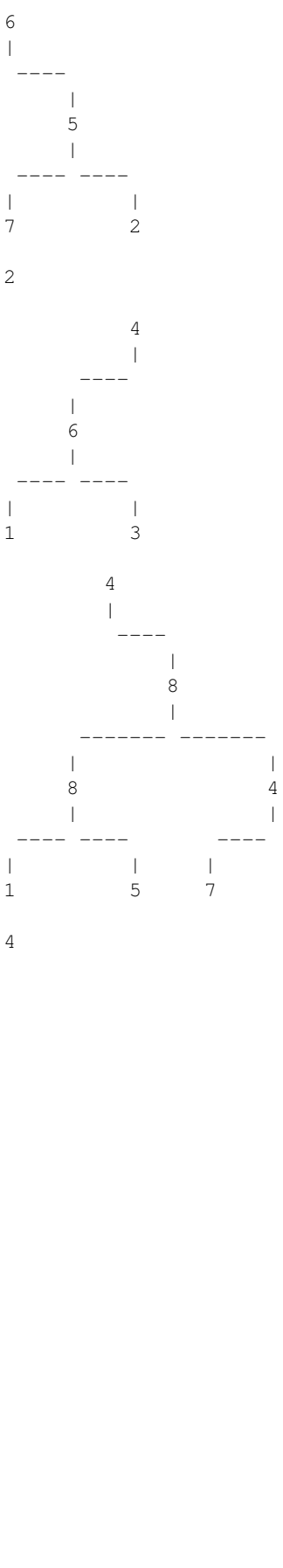
La primera linia de l'entrada descriu el format en el que es descriuen els arbres, o bé INLINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

Sortida

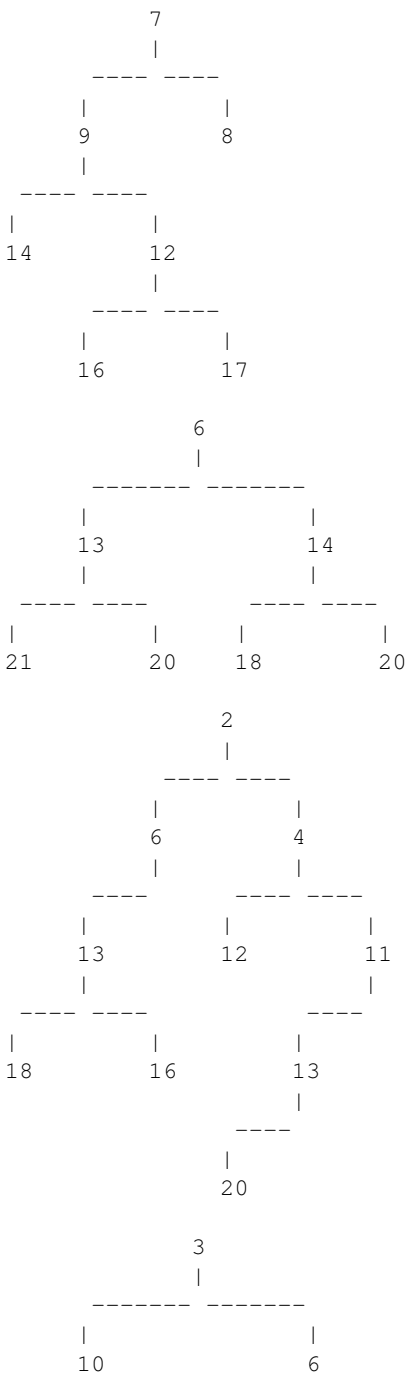
Per a cada cas, la sortida conté el corresponent arbre de sumes d'ancestres. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta sortida. Només cal que implementeu la funció abans esmentada.

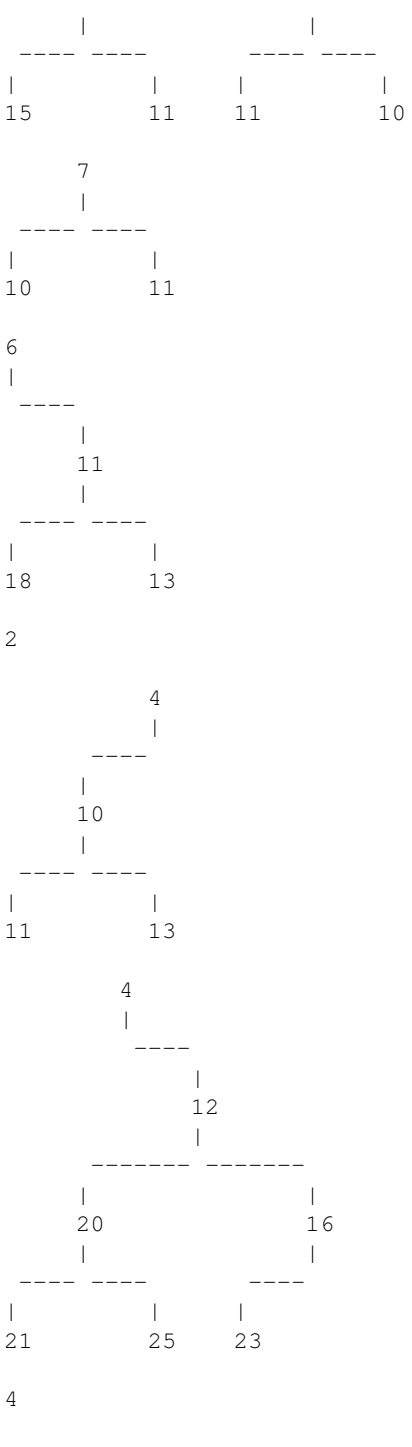
Exemple d'entrada 1





Exemple de sortida 1





Exemple d'entrada 2

```
INLINEFORMAT
7(2(5,3(4,5)),1)
6(7(8,7),8(4,6))
2(4(7(5,3),),2(8,7(2(7,),)))
3(7(5,1),3(5,4))
7(3,4)
6(,5(7,2))
2
4(6(1,3),)
```

```
4(,8(8(1,5),4(7,)))
4
```

Exemple de sortida 2

```
7(9(14,12(16,17)),8)
6(13(21,20),14(18,20))
2(6(13(18,16),),4(12,11(13(20,),)))
3(10(15,11),6(11,10))
```

```
7(10,11)
6(,11(18,13))
2
4(10(11,13),)
4(,12(20(21,25),16(23,)))
4
```

Observació

Les vostres funcions i subfuncions que creeu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema. Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autoria: PRO2

Generació: 2026-01-25T16:51:39.216Z