Arbre amb inordre i arbre amb postordre — X97696_ca

Nota: En aquest exercici, les entrades contenen àrbres binaris d'enters. Els valors dels nodes d'aquests àrbres d'entrada no importen, només importa l'estructura dels àrbres. Per facilitar la llegibilitat dels exemples, totes les entrades seran àrbres amb només 0's als nodes, cosa que, com hem comentat, no és rellevant.

Preliminars

Recordeu que el recorregut en inordre d'un arbre és la llista dels nodes de l'arbre ordenada com segueix: en primer lloc, el recorregut en inordre del fill esquerra de l'arbre, després l'arrel de l'arbre, i després el recorregut en inordre del fill dret de l'arbre. En altres paraules:

- $Inordre(x(t_1, t_2)) = Inordre(t_1) \cdot x \cdot Inordre(t_2)$
- $Inordre(()) = ()$, és a dir, l'inordre de l'arbre buit és l'arbre buit.

Recordeu també que el recorregut en postordre d'un arbre és la llista dels nodes de l'arbre ordenada com segueix: en primer lloc, el recorregut en postordre del fill esquerra de l'arbre, després el recorregut en postordre del fill dret de l'arbre, i finalment l'arrel de l'arbre. En altres paraules:

- $Postordre(x(t_1, t_2)) = Postordre(t_1) \cdot Postordre(t_2) \cdot x$
- $Postordre(()) = ()$, és a dir, el postordre de l'arbre buit és l'arbre buit.

Exercici:

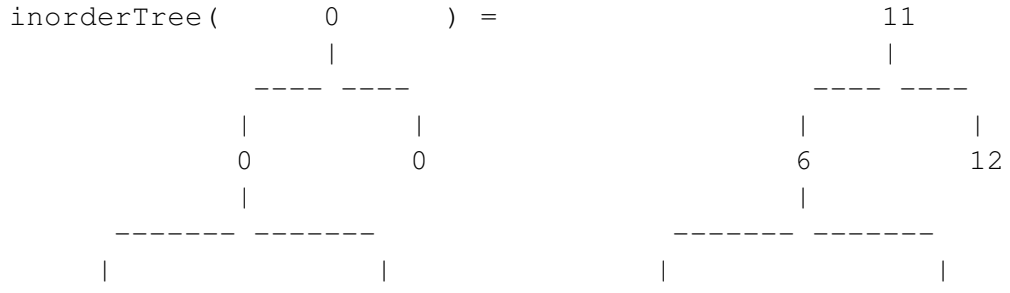
Haureu d'implementar dues funcions **RECURSIVES**.

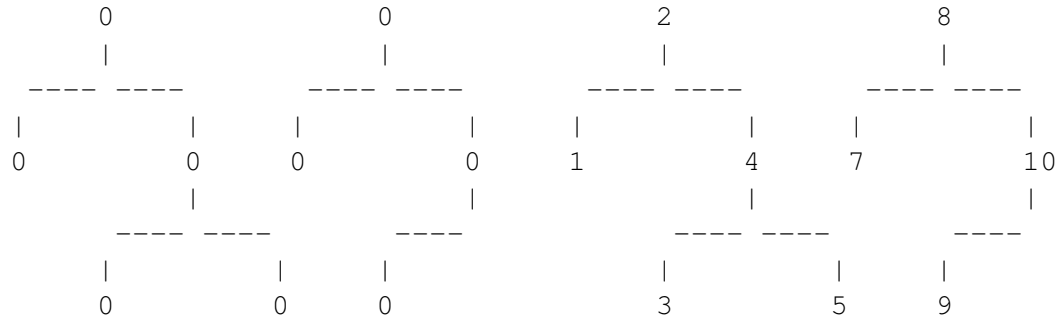
La primera funció que heu d'implementar rep un arbre binari d'enters i ha de retornar un altre arbre binari d'enters, amb exactament la mateixa estructura (conjunt de posicions) que el que s'ha rebut d'entrada, i a on cada node guardarà la posició d'aquell node en el recorregut en inordre de l'arbre.

```
// Pre:  Sigui T el valor inicial de l'arbre t que es rep com a paràmetre.
// Post: Sigui T' l'arbre retornat. T i T' tenen exactament la mateixa estructu
//       Sigui n1,n2,...,nk els nodes de T' en el recorregut en inordre de T'.
//       Llavors, n1 guarda el valor 1, n2 guarda el valor 2, ..., nk guarda el
BinTree<int> inorderTree(BinTree<int> t);
```

Aquí tenim un exemple de comportament de la funció:

```
inorderTree(0(0(0(0,0(0,0)),0(0,0(0,))),0)) = 11(6(2(1,4(3,5)),8(7,10(9,))),12)
```



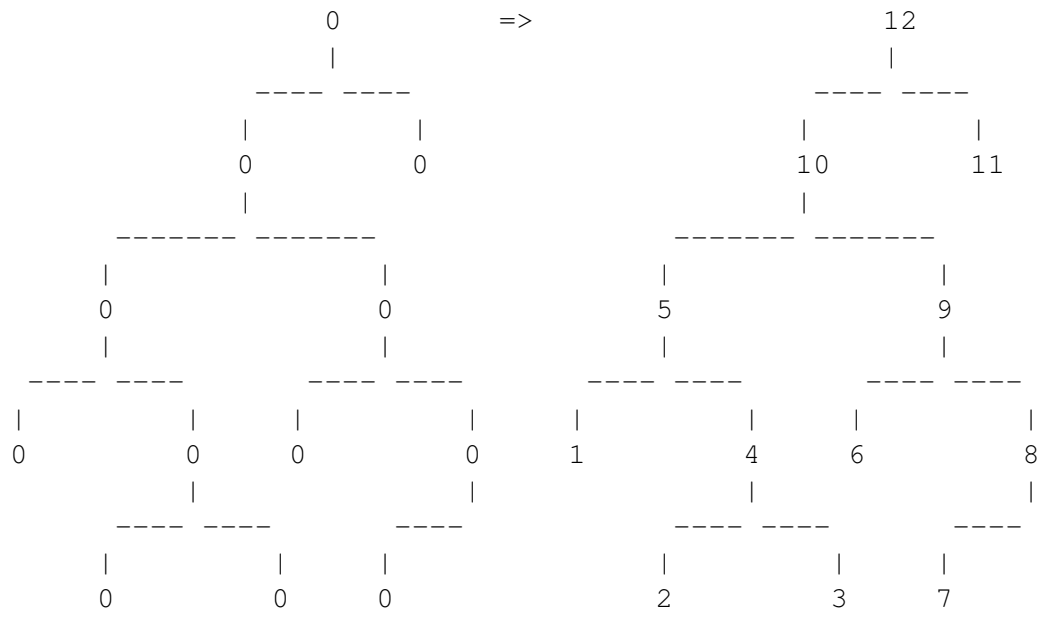


La segona funció que heu d'implementar rep un arbre binari d'enters i ha de retornar un altre arbre binari d'enters, amb exactament la mateixa estructura (conjunt de posicions) que el que s'ha rebut d'entrada, i a on cada node guardarà la posició d'aquell node en el recorregut en postordre de l'arbre.

```
// Pre:  Sigui T el valor inicial de l'arbre t que es rep com a paràmetre.
// Post: Sigui T' l'arbre retornat. T i T' tenen exactament la mateixa estructu
//       Sigui n1,n2,...,nk els nodes de T' en el recorregut en postordre de T'
//       Llavors, n1 guarda el valor 1, n2 guarda el valor 2, ..., nk guarda el
BinTree<int> postorderTree(BinTree<int> t);
```

Aquí tenim un exemple de comportament de la funció:

```
postorderTree(0(0(0(0,0(0,0)),0(0,0(0,))),0)) = 12(10(5(1,4(2,3)),9(6,8(7,))),1
```



Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc, BinTree.hh, inorderANDpostorderTree.hh`. Només cal que creeu `inorderANDpostorderTree.cc`, posant-hi els includes que calguin i implementant les funcions `inorderTree` i `postorderTree`. Només cal que pugeu `inorderANDpostorderTree.c` al jutge.

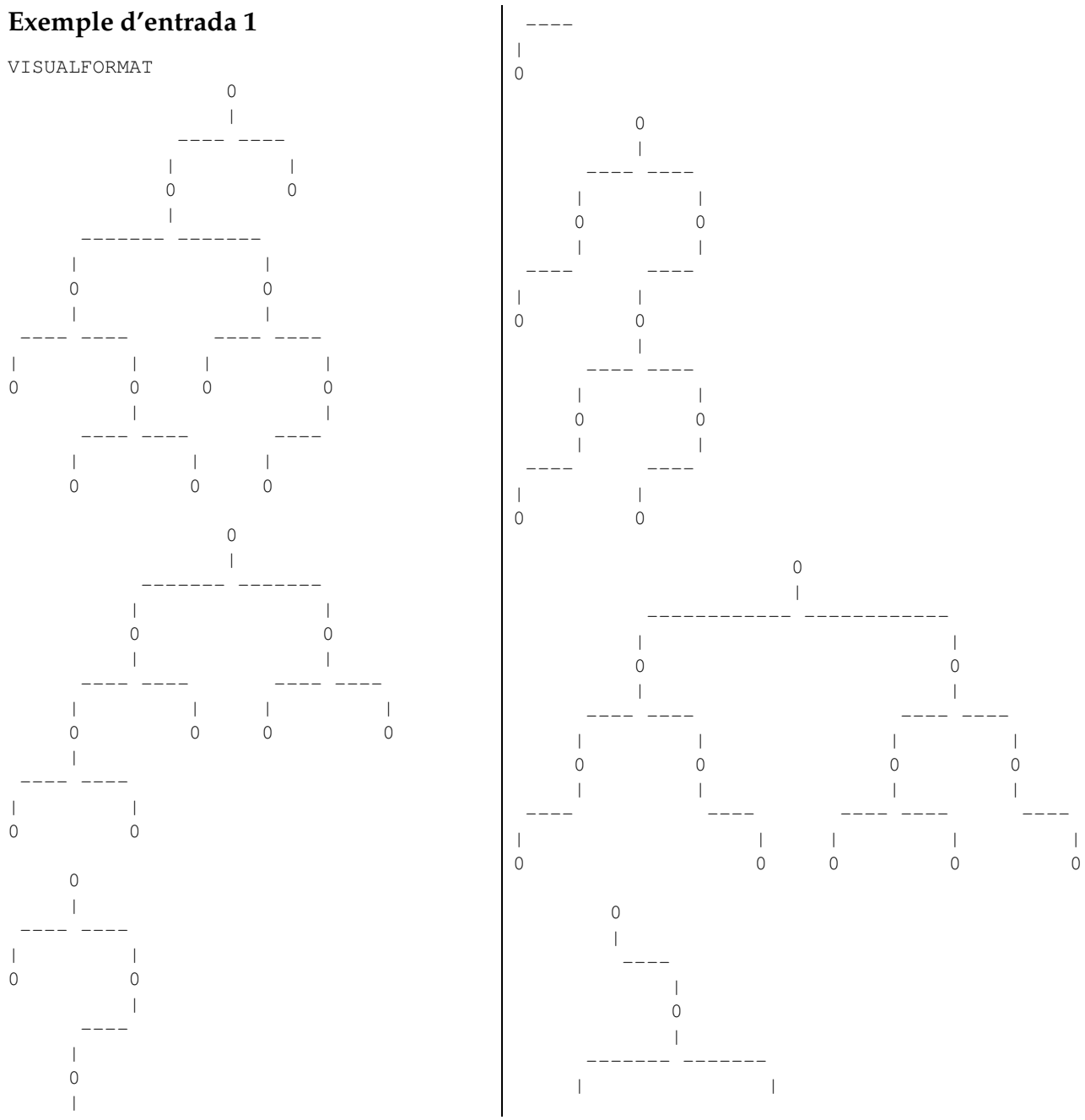
Entrada

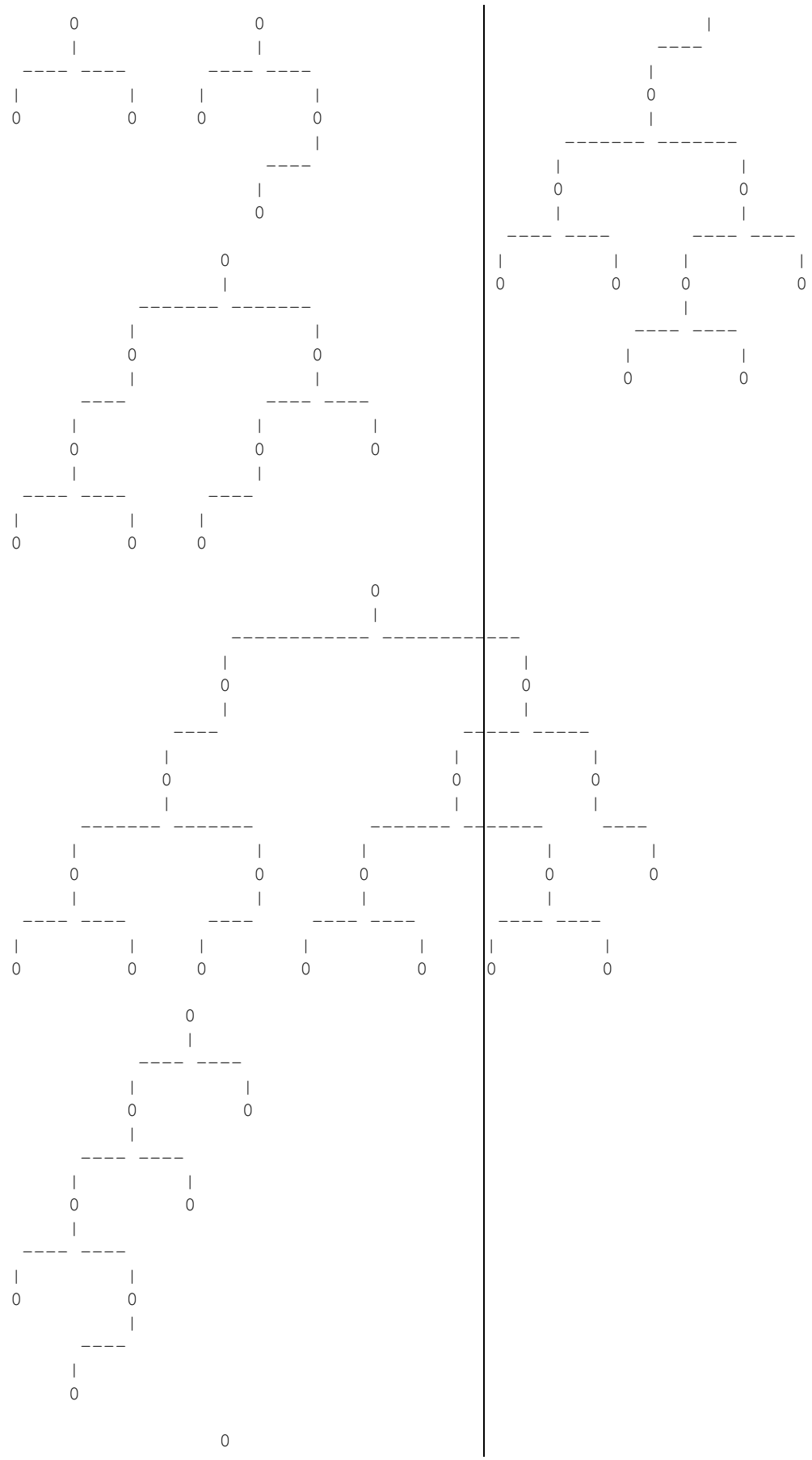
La primera linia de l'entrada descriu el format en el que es descriuen els arbres, o bé INLINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari d'enters (amb només el valor 0 als nodes, tot i que això és irrellevant). Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu les funcions abans esmentades.

Sortida

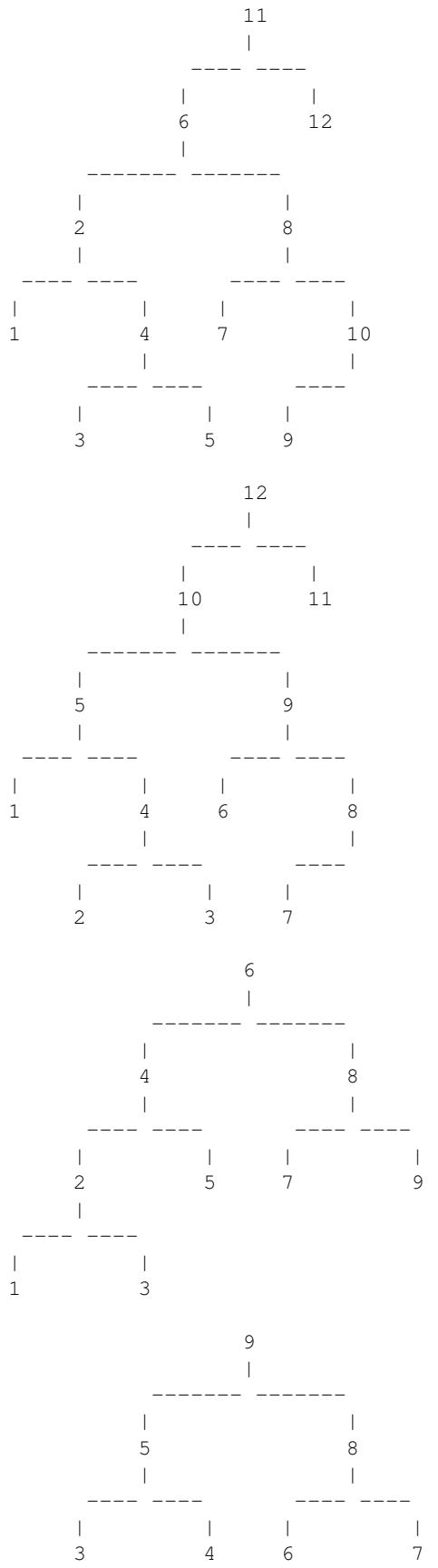
Per a cada cas, cal escriure els dos arbres binaris resultants de cridar a les funcions abans esmentades amb l'arbre d'entrada. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta sortida. Només cal que implementeu les funcions abans esmentades.

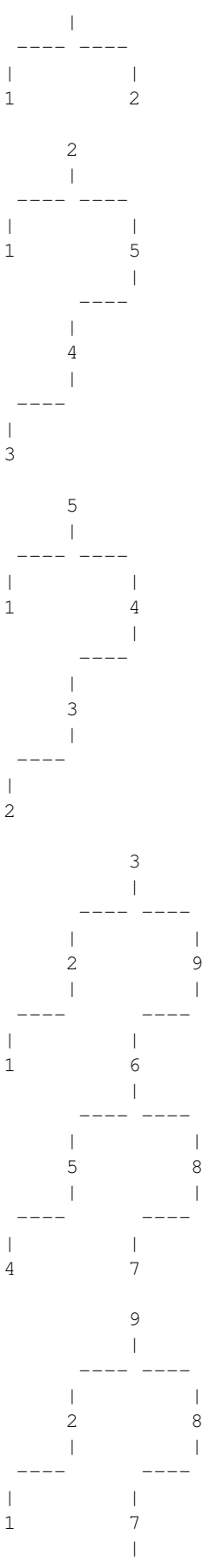
Exemple d'entrada 1

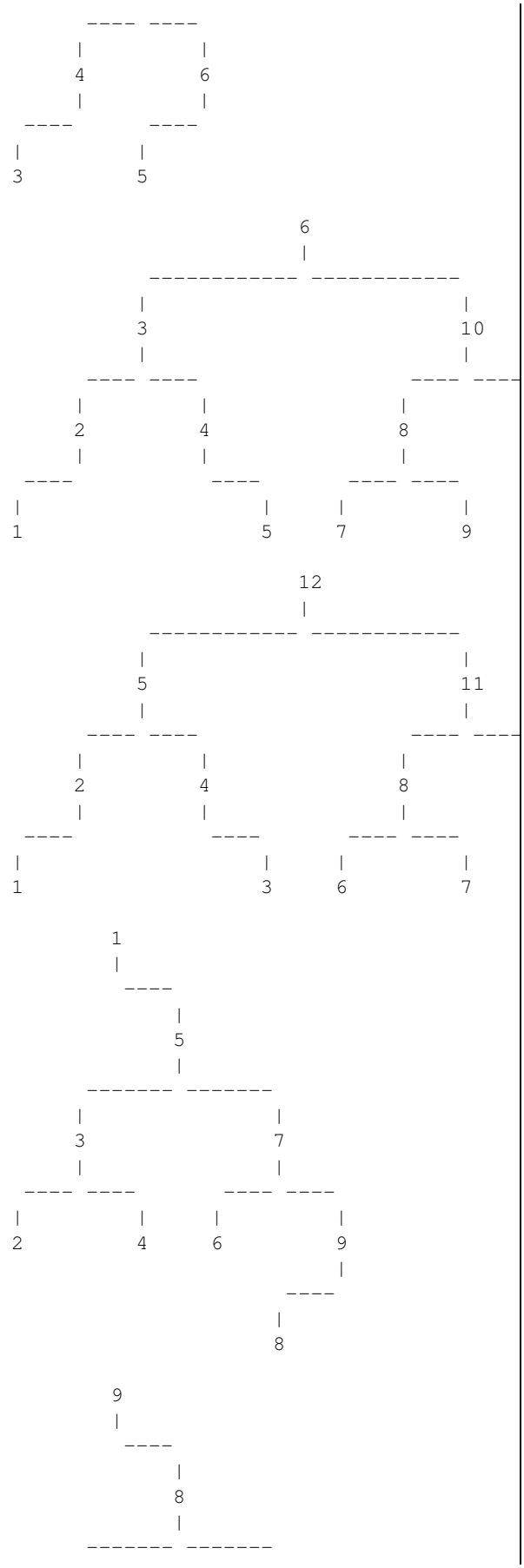


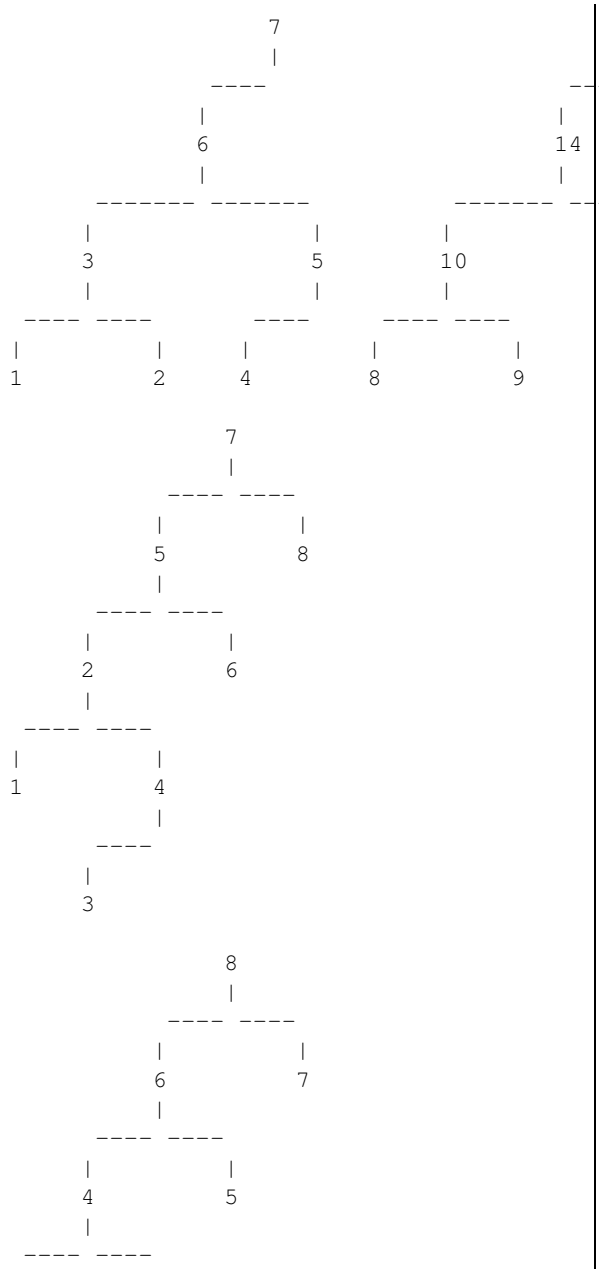


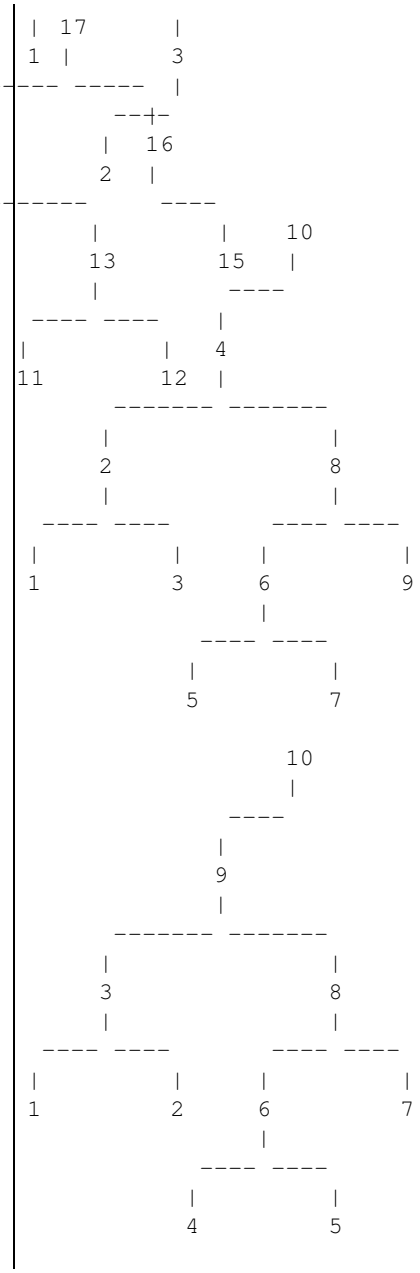
Exemple de sortida 1











Exemple d'entrada 2

```
INLINEFORMAT
0(0(0(0,0(0,0)),0(0,0(0,))),0)
0(0(0(0,0),0),0(0,0))
0(0,0(0(0,),))
0(0(0,),0(0(0(0,),0(0,)),))
0(0(0(0,),0(,0)),0(0(0,0),0(,0)))
0(,0(0(0,0),0(0,0(0,))))
0(0(0(0,0),),0(0(0,),0))
0(0(0(0(0,0),0(0,)),),0(0(0(0,0),0(0,0)),0(0,0)))
0(0(0(0,0(0,)),0),0)
0(0(0(0,0),0(0(0,0),0)),)
```

Exemple de sortida 2

```
11(6(2(1,4(3,5)),8(7,10(9,))),12)
12(10(5(1,4(2,3)),9(6,8(7,))),11)
6(4(2(1,3),5),8(7,9))
9(5(3(1,2),4),8(6,7))
2(1,5(4(3,),))
5(1,4(3(2,),))
3(2(1,),9(6(5(4,),8(7,)),))
9(2(1,),8(7(4(3,),6(5,)),))
6(3(2(1,),4(,5)),10(8(7,9),11(,12)))
12(5(2(1,),4(,3)),11(8(6,7),10(,9)))
1(,5(3(2,4),7(6,9(8,))))
9(,8(3(1,2),7(4,6(5,))))
5(4(2(1,3),),8(7(6,),9))
9(4(3(1,2),),8(6(5,),7))
8(7(4(2(1,3),6(5,)),),16(12(10(9,11),14(13,15)),17(,18
```

```
18(7(6(3(1,2),5(4,)),),17(14(10(8,9),13(11,12)),16(15,)))
7(5(2(1,4(3,)),6),8)
8(6(4(1,3(2,)),5),7)
```

```
10(4(2(1,3),8(6(5,7),9)),)
10(9(3(1,2),8(6(4,5),7)),)
```

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

Autor : PRO2
Generació : 2024-04-05 07:54:04