

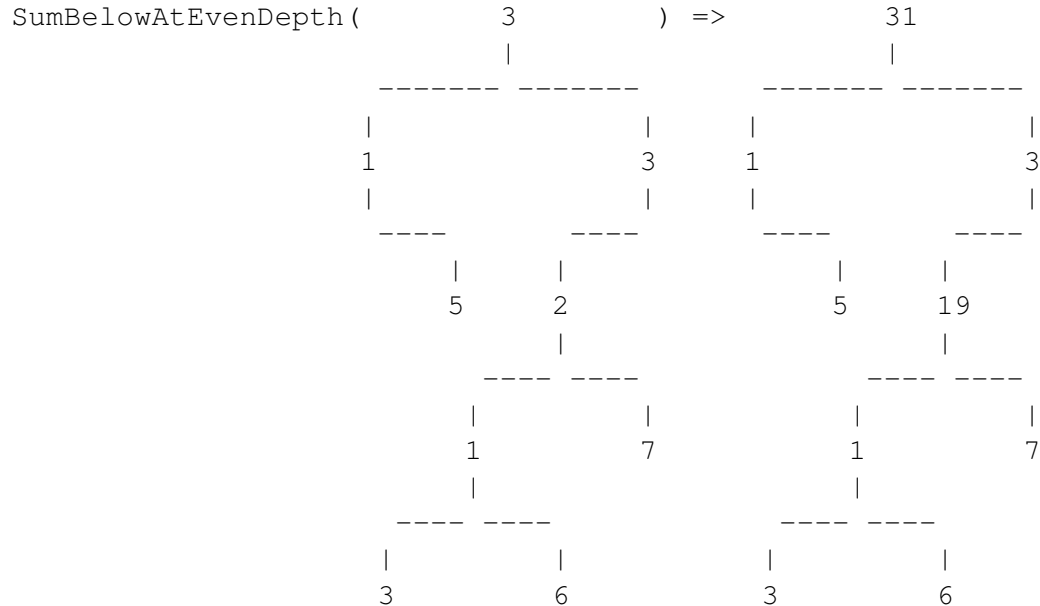
**Reemplaça els valors dels nodes a profunditat parell en un arbre per la suma per sota** **X68048\_ca**

Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna un nou arbre amb la mateixa estructura, i a on cada posició a profunditat parell conté la suma de nodes del subarbre que penja d'aquella mateixa posició a l'arbre inicial, i a cada posició a profunditat senar hi ha exactament el mateix valor que es troba en aquella posició a l'arbre inicial.

Sobreentenem que l'arrel de l'arbre està a profunditat 0, els nodes directes des de l'arrel són a profunditat 1, els nodes a distància dos de l'arrel són a profunditat 2, i així successivament. Aquesta és la capelera:

```
// Pre: Sigui T el valor inicial de t.
// Post: Retorna un arbre d'enters R amb la mateixa estructura que T.
//       Per a cada posició p de T i R, si p és a profunditat senar,
//       llavors T i R tenen el mateix valor a posició p.
//       En canvi, si p es a profunditat parell, llavors el valor de R a posició
//       p és la suma de tots els valors que es troben a T a posició p o per sota.
BinTree<int> SumBelowAtEvenDepth(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`, `BinaryTree.hh`, `SumBelowAtEvenDepth.hh`. Us falta crear el fitxer `SumBelowAtEvenDepth.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Només cal que pugeu `SumBelowAtEvenDepth.cc` al jutge.

## Entrada

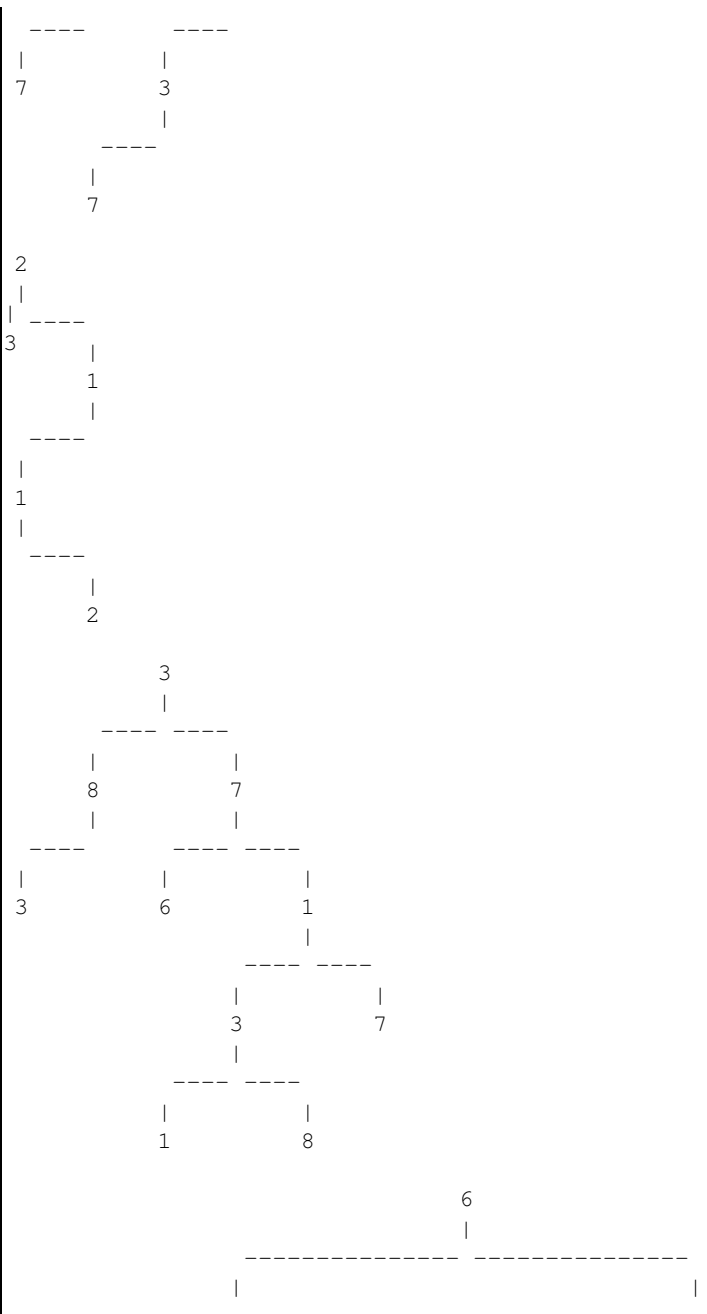
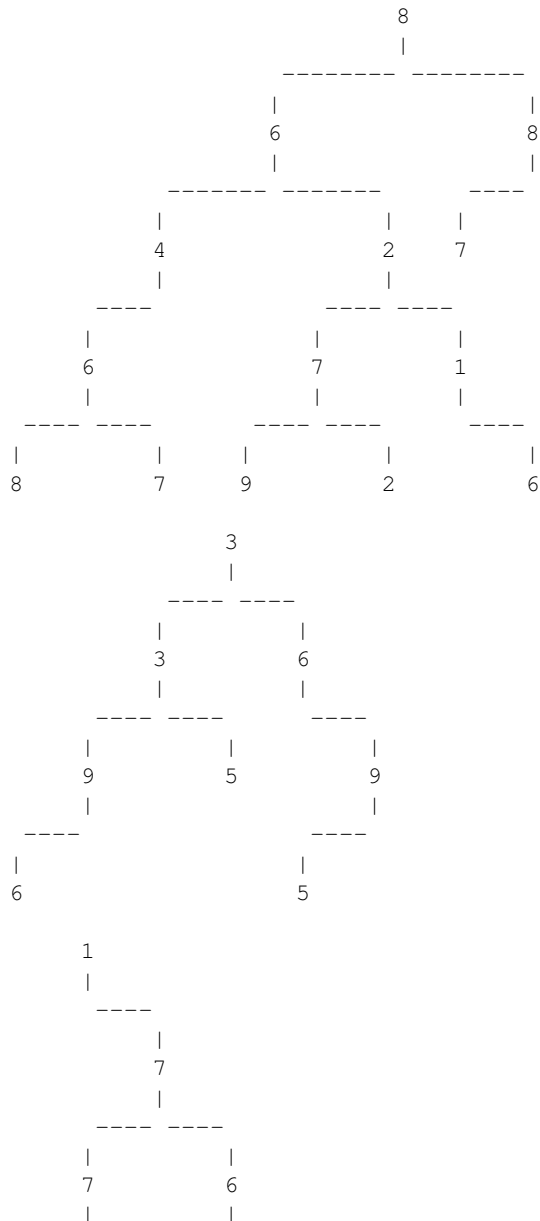
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé IN-LINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega de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

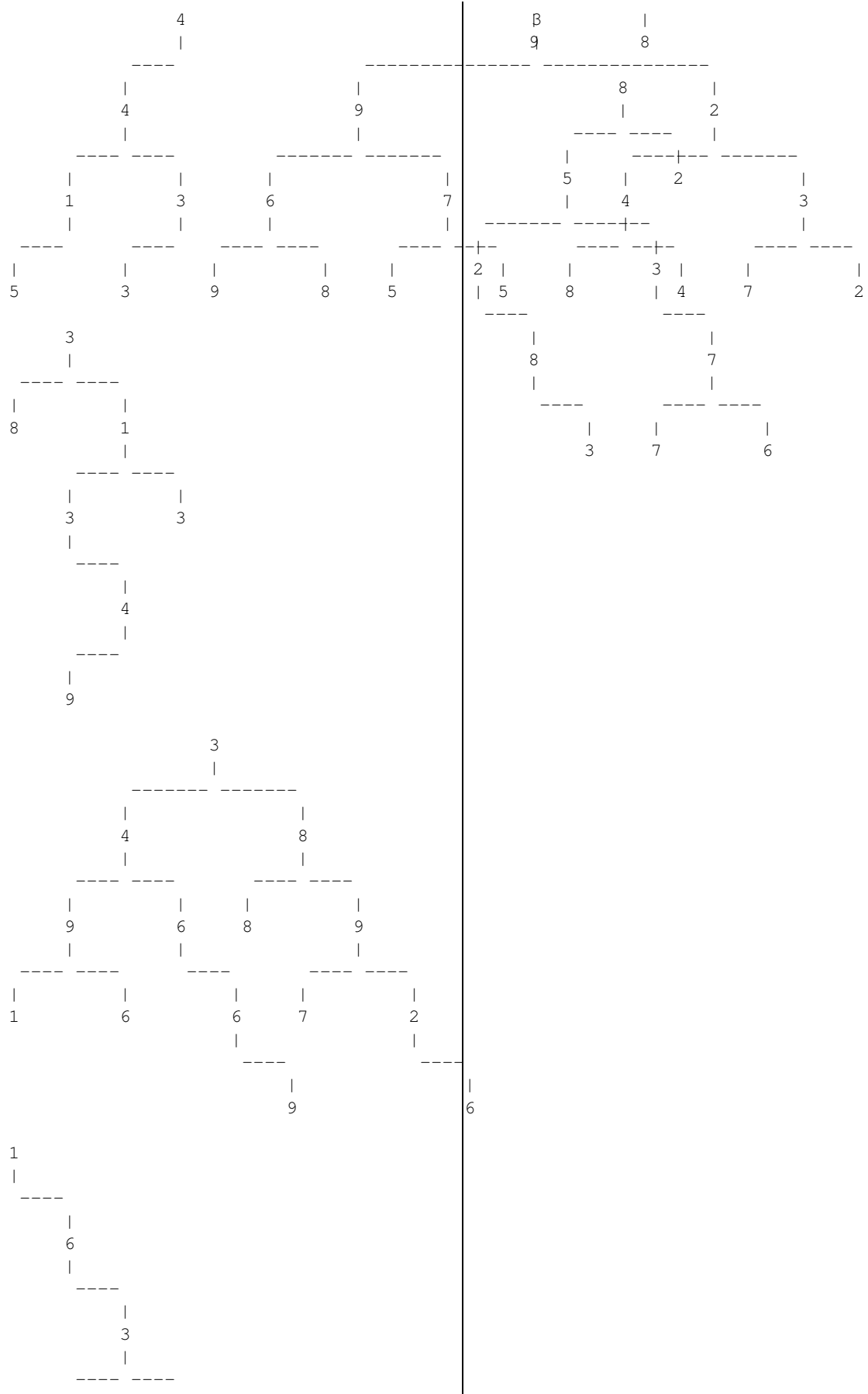
## Sortida

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

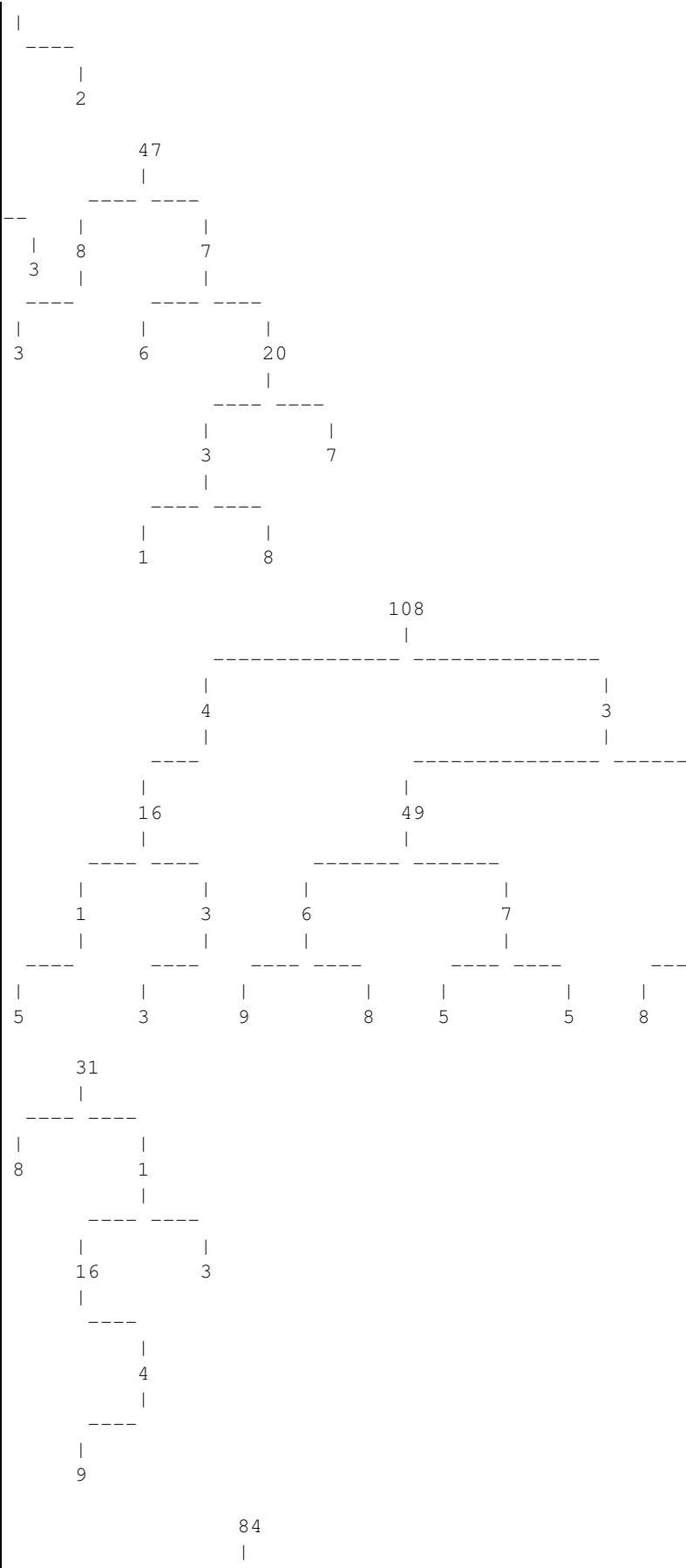
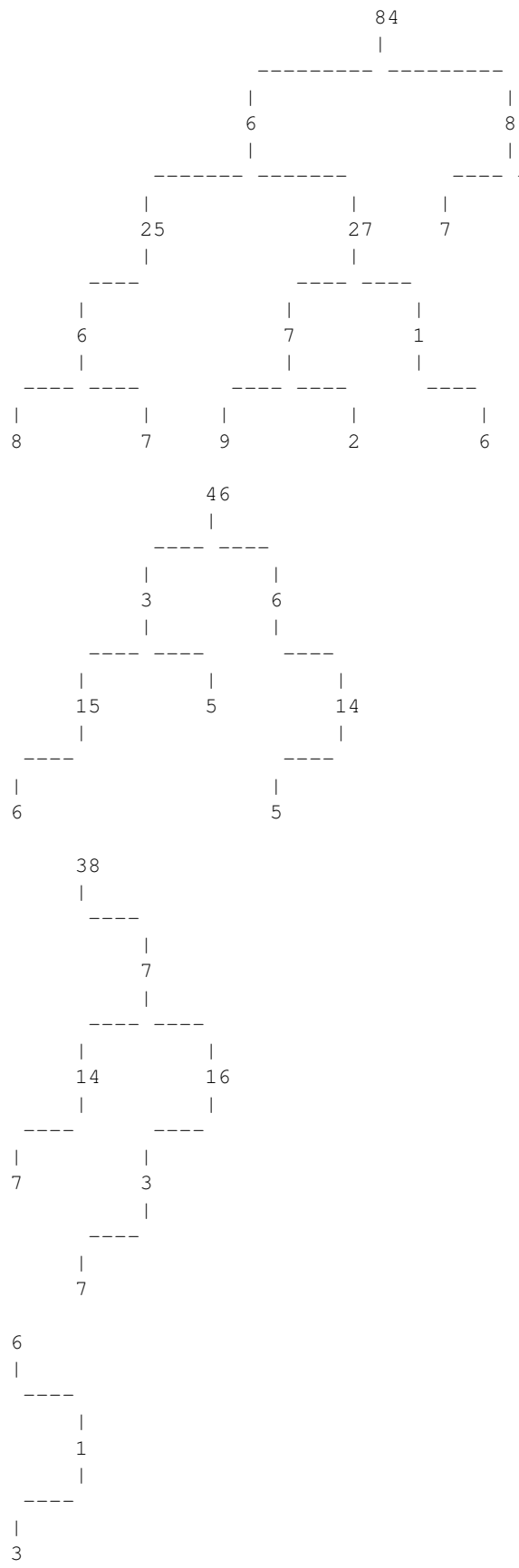
### Exemple d'entrada 1

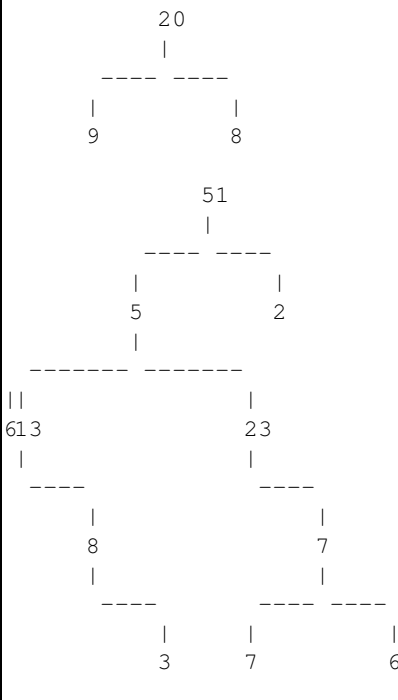
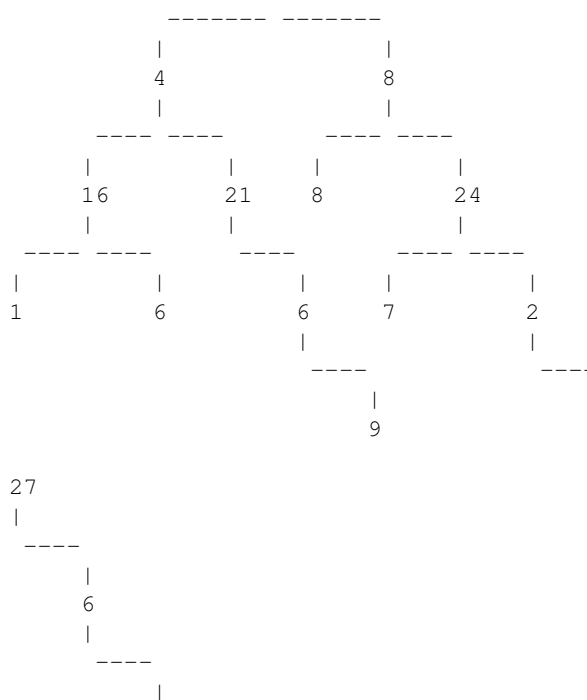
VISUALFORMAT





# Exemple de sortida 1





### Exemple d'entrada 2

```

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

```

### Exemple de sortida 2

```

84(6(25(6(8,7),),27(7(9,2),1(,6))),8(7,3))
46(3(15(6,),5),6(,14(5,)))
38(,7(14(7,),16(3(7,),)))
6(,1(3(,2),))
47(8(3,),7(6,20(3(1,8),7)))
108(4(16(1(5,),3(3,)),),3(49(6(9,8),7(5,5)),30(4(8,4),
31(8,4),16(7,2(9,))),3))
84(4(16(1,6),21(,6(,9))),8(8,24(7,2(,6))))
27(,6(,20(9,8)))
51(5(13(,8(,3)),23(,7(7,6))),2)

```

### Observació

La vostra funció i subfuncions que creu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema.

### Informació del problema

Autor : PRO2  
 Generació : 2023-10-21 13:51:14

© Jutge.org, 2006–2023.  
<https://jutge.org>