

(BinTree) Nombre de fulles amb mateix valor que l'arrelX72420_ca

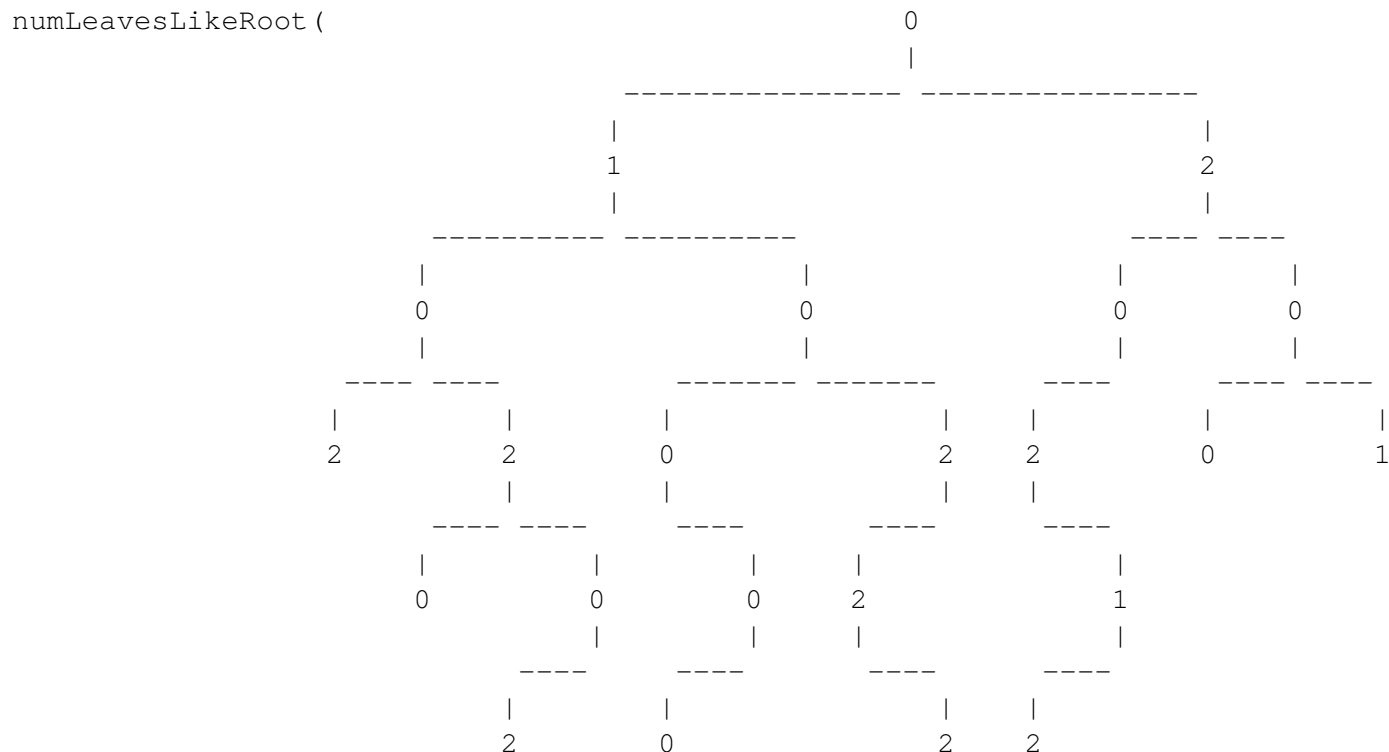
Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters no-buit, retorna el nombre de fulles que tenen el mateix valor que l'arrel.

Aquesta és la capçalera:

```
// Pre: t és no buit.
// Post: Retorna el nombre de nodes de t que tenen el mateix valor que l'arrel.
int numLeavesLikeRoot(BinTree<int> t);
```

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

```
numLeavesLikeRoot ( 0 (1 (0 (2, 2 (0, 0 (2, ))), 0 (0 (, 0 (0, ))), 2 (2 (, 2, ))), 2 (0 (2 (, 1 (2, ))), ,
```



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

Entrada

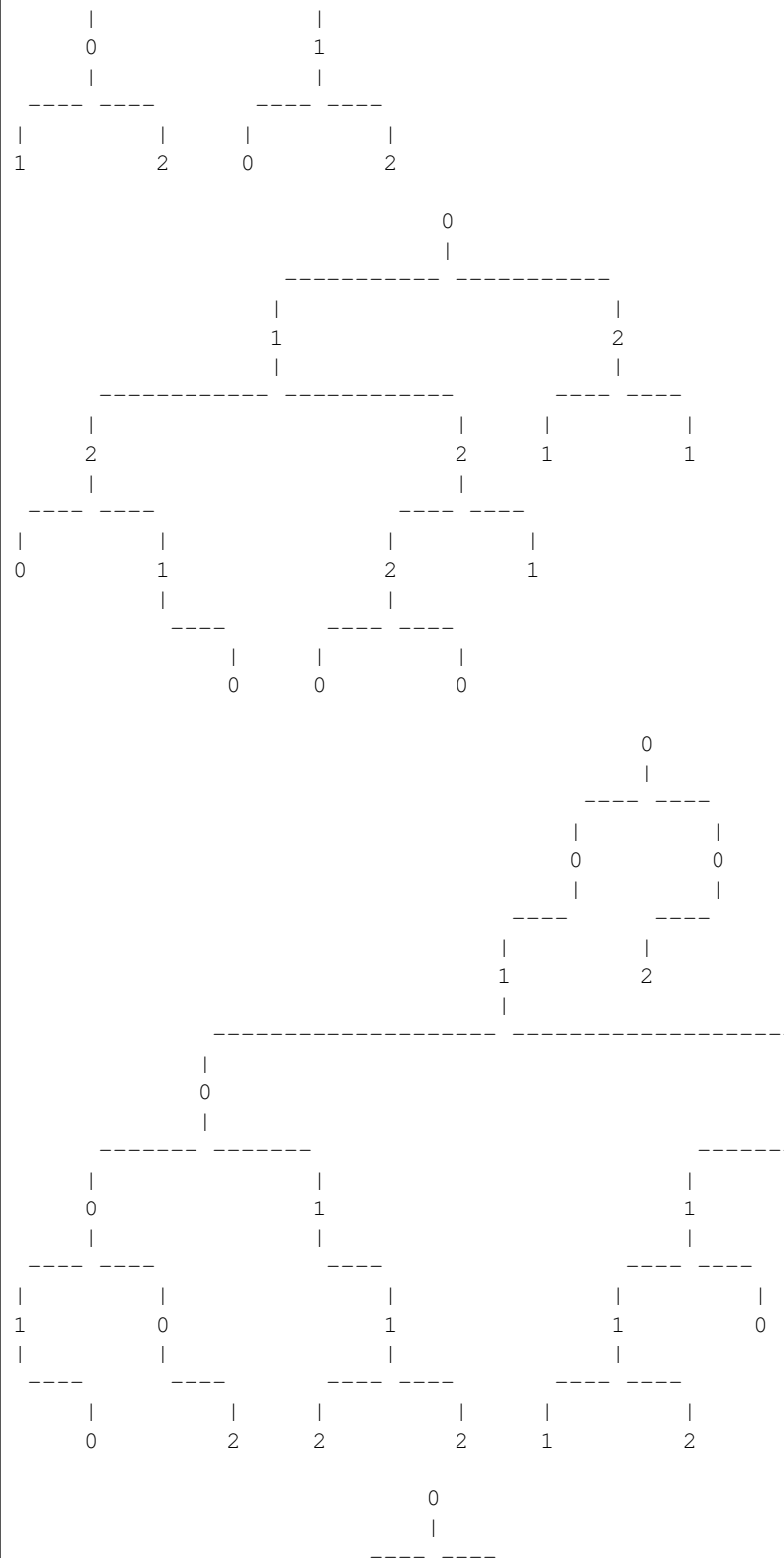
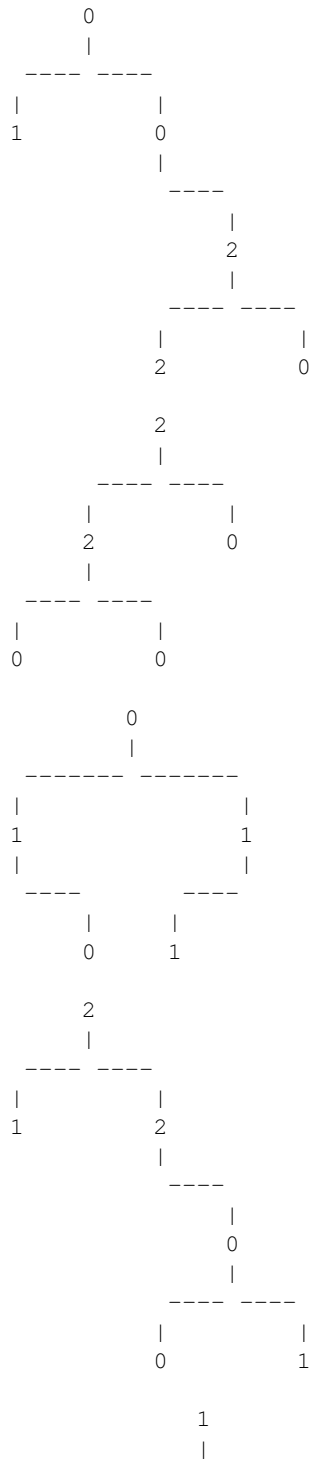
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `INLINEFORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega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

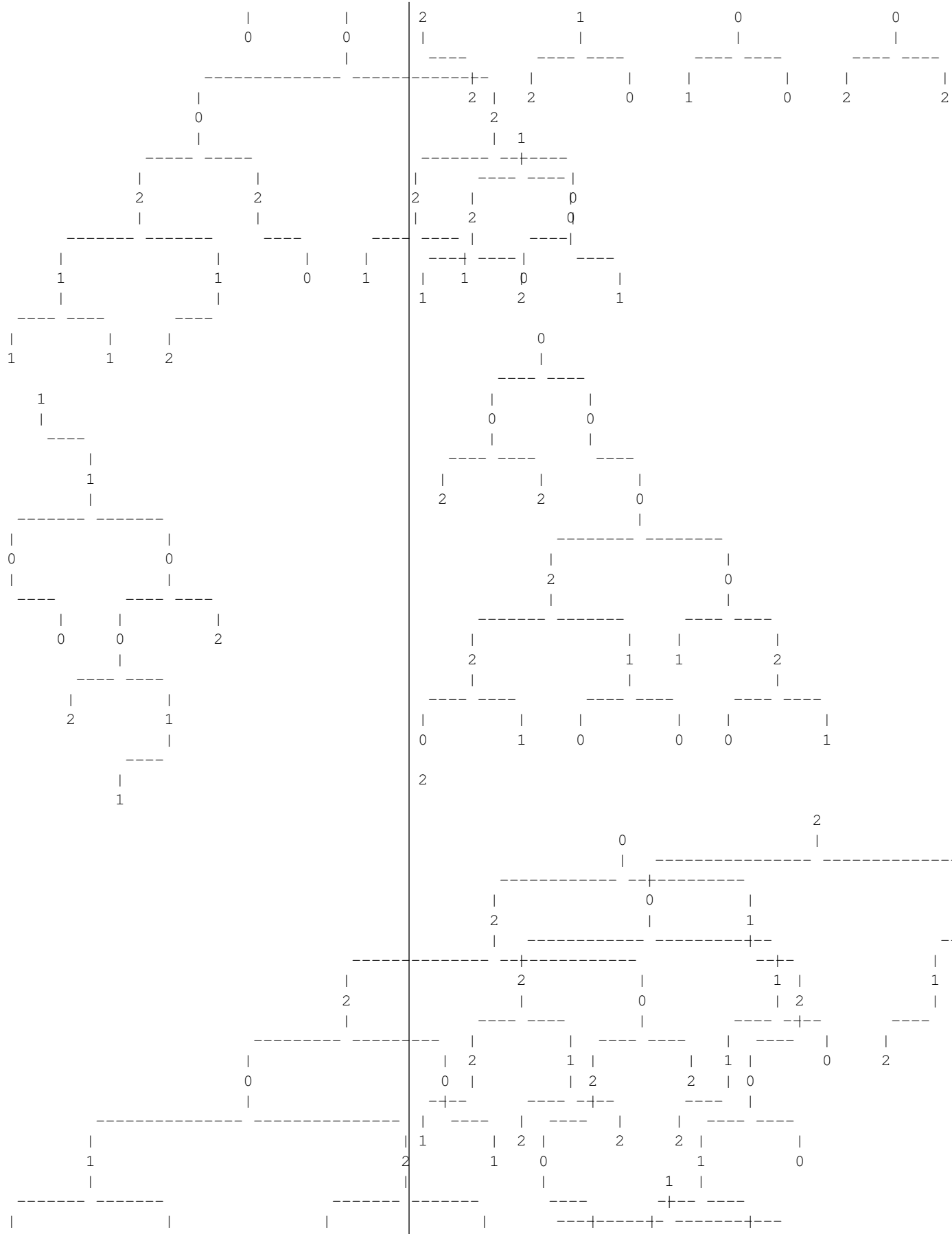
Sortida

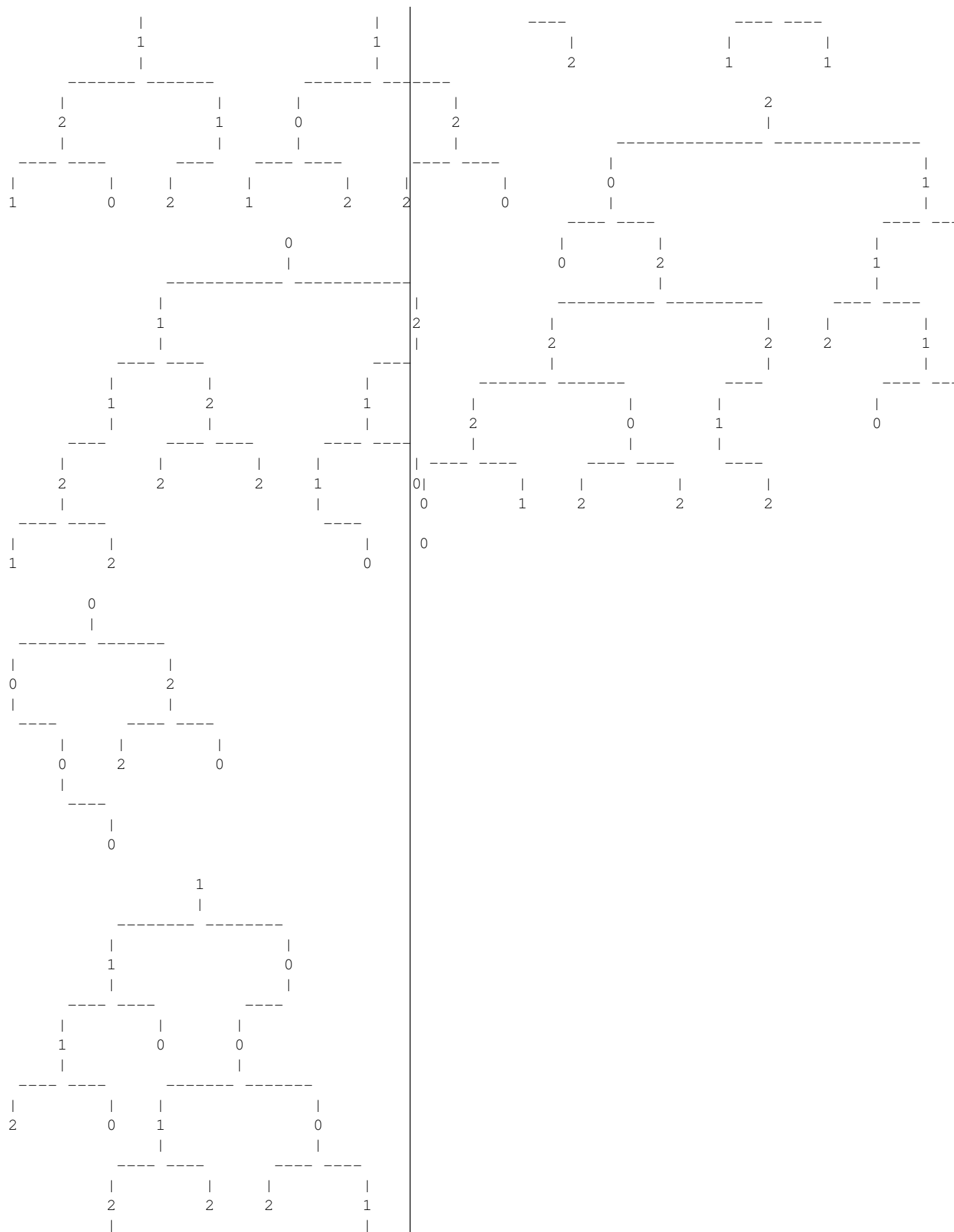
Per a cada cas, la sortida conté el corresponent resultat de la funció. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquest resultat. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

VISUALFORMAT







1
0
1
0
1
4
2
3
1

5
2
4
1
5
2
2
2
2
4
1

INLINEFORMAT

0(1(0(,2(2,0)))	0
2(2(0,0),0)	1
0(1(,0),1(1,))	0
2(1,2(,0(0,1)))	1
1(0(1,2),1(0,2))	4
0(1(2(0,1(,0)),2(2(0,0),1)),2(1,1))	2
0(0(1(0(0(1(,0),0(,2)),1(,1(2,2))),1(1(1(1(2,0),1(1,0(,2))))),),0(2,))	13
0(0,0(0(2(1(1,1),1(2,)),2(,0)),2(2(1,1),0(,)))	5
1(,1(0(,0),0(0(2,1(1,)),2)))	1
0(2(2(0(1(2(,2),1(2,0)),2(0(1,0),0(2,2))),2(,1)),0(2(0(,1(,1)),),2)),1	2
1(2(1,2),0(,1))	4
0(0(2,2),0(,0(2(2(0,1),1(0,0)),0(1,2(0,1))))	1
2	5
2(0(2(2(1,),1(2,2)),1(1(2,),0)),0(1(2,),2))	2
1(1(2(1,0),1(2,)),1(0(1,2),2(2,0)))	2
0(1(1(2(1,2),),2(2,2)),2(1(1(,0),0),))	2
0(0(,0(,0)),2(2,0))	2
1(1(1(2,0),0),0(0(1(2(,2),2),0(2,1(1,1))))	4
2(0(0,2(2(2(0,1),0(2,2)),2(1(,2),))),1(1(1(1(0,0)),0))	21
0	0

1
0
1
0
1
4
2
13
(10
5
, 20
4
) 1

$$\begin{aligned} & 2(, 1)), 0(2(0(, 1(, 1)),), 2)), 1(, 2(0(1(0, 0(2, 0)), 0),)))) \\ & 4 \\ &)1)) \\ & 5 \\ &)2 \\ & 2 \\ & 2 \\ & 2 \\ & , 4) \\ & 21(1(0, 0)), 0)) \end{aligned}$$

La vostra funció 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.

Autoria: PRO2

Generació: 2026-01-25T21:24:03.266Z

© Jutge.org, 2006–2026.
<https://jutge.org>