

Nombre de nodes amb valor estrictament major que el valor del seu node pare **X31410_ca**

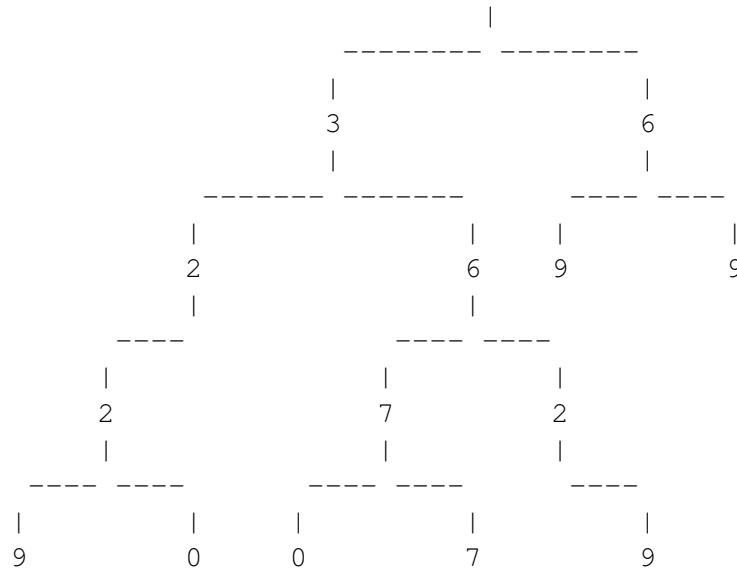
Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna el nombre de nodes que guarden un valor estrictament major que el valor guardat en el seu node pare. Fixeu-vos que l'arrel de l'arbre no compta perquè no té node pare. Aquesta és la capçalera:

```
// Pre:
// Post: Retorna el nombre de nodes de t que no son l'arrel de t i que guarden
//        estrictament major que el valor guardat al seu node pare.
int numIncreasing(BinTree<int> t);
```

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

```
numIncreasing( 6(3(2(2(9,0),),6(7(0,7),2(,9))),6(9,9)) ) = 6
```

```
numIncreasing(                                     6                                     ) = 6
```



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

Entrada

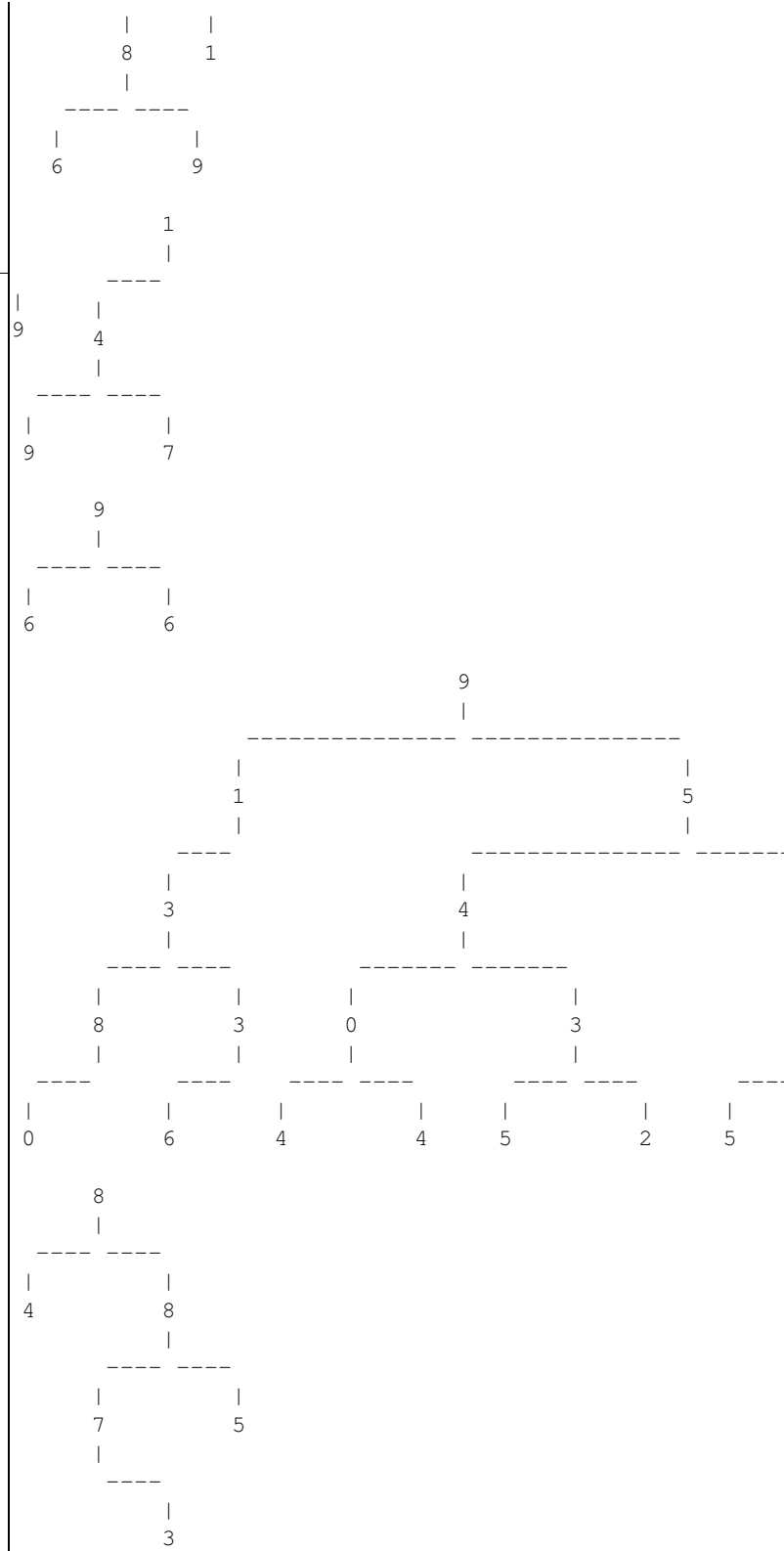
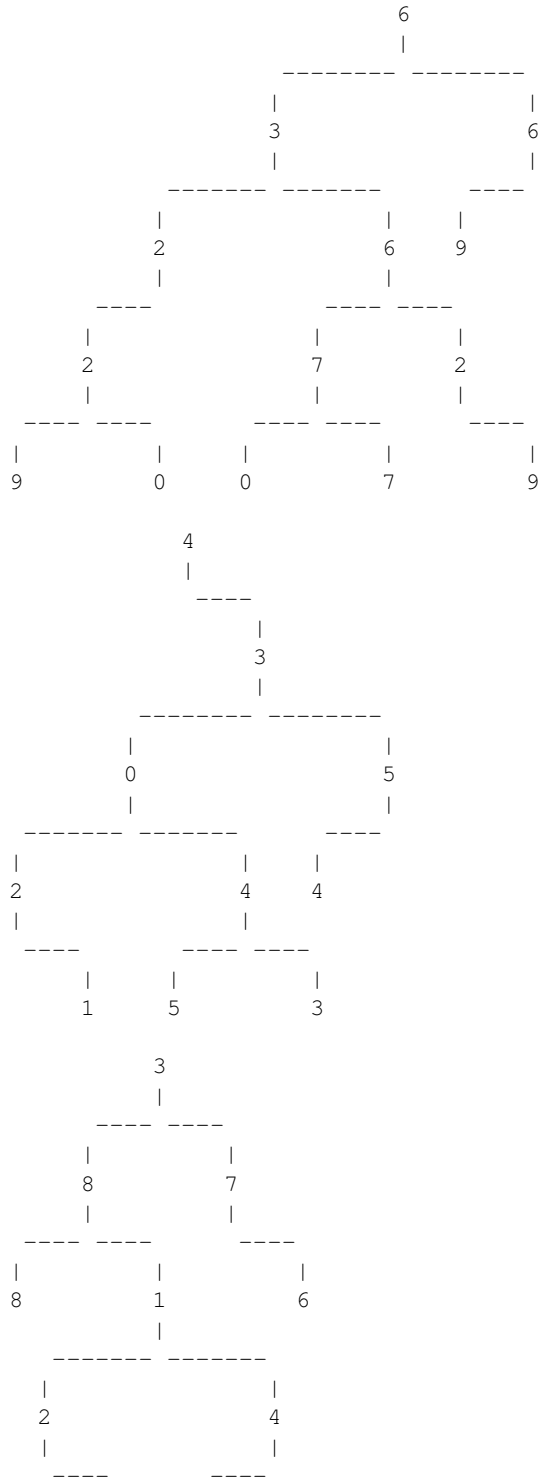
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `IN-LINEFORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega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

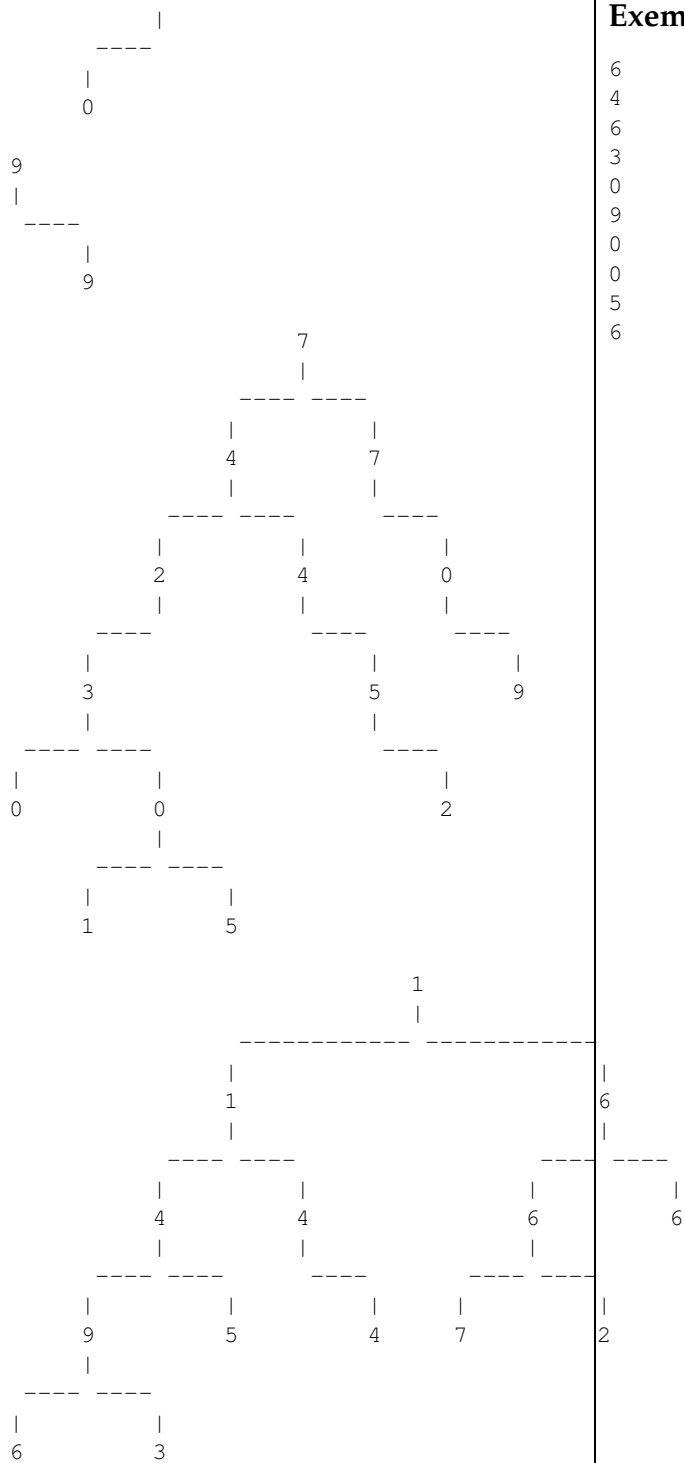
Sortida

Per a cada cas, la sortida conté el corresponent 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





Exemple de sortida 1

6
4
6
3
0
9
0
0
5
6

Exemple d'entrada 2

INLINEFORMAT

6(3(2(2(9,0),),6(7(0,7),2(,9))),6(9,9))

4(,3(0(2(,1),4(5,3)),5(4,)))

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

1(4(9,7),)

9(6,6)

9(1(3(8(0,),3(6,)),),5(4(0(4,4),3(5,2))),8(4(5,5),8(8,4)))

8(4,8(7(,3(0,)),5))

9(,9)

7(4(2(3(0,0(1,5))),),4(,5(,2))),7(,0(,9)))

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

Exemple de sortida 2

6		0
4		9
6		0
3		0
		5
		6

Observació

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.

Informació del problema

Autor : PRO2

Generació : 2024-02-22 19:58:54

© *Jutge.org*, 2006–2024.

<https://jutge.org>