

---

## Nombre de fulles d'un arbre

X12047\_ca

---

Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna el nombre de fulles que té. Això significa que retorna el nombre de nodes de l'arbre tals que els seus dos fills son buits. Aquesta és la capçelera:

```
// Pre:  
// Post: Retorna el nombre de fulles de t  
int numberOfLeaves(BinTree<int> t);
```

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

```
t:          3  
           |  
    -----  
   |               |  
  1               4  
   |               |  
-----  
|       |       |  
2       5       1
```

=>

3

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

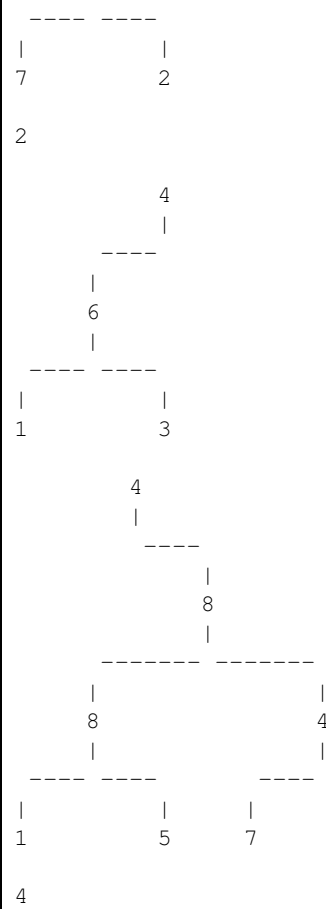
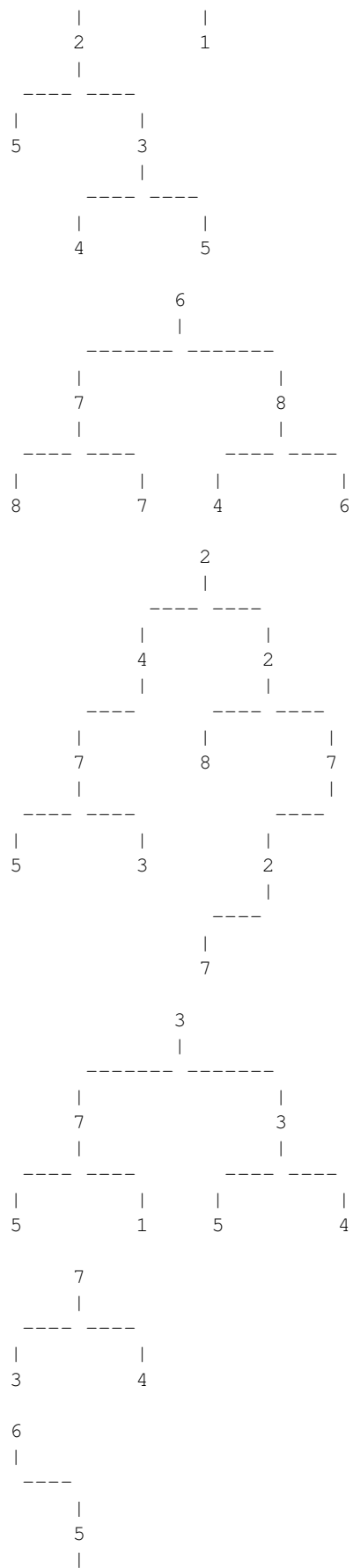
### Sortida

Per a cada cas, la sortida conté el corresponent nombre de fulles de l'arbre. 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

```
7  
|  
-----
```



## Exemple de sortida 1

4  
4  
4  
4

2  
2  
1  
2  
3  
1

## Exemple d'entrada 2

```
INLINEFORMAT
0 (55 (29, -47 (-15, 98)), -18)
-94 (82 (-21, 80), -16 (63, -85))
-27 (-50 (6 (13, -56), ), 23 (2, 36 (-2 (-37, ), )))
-56 (-5 (-100, -37), 7 (-70, -18))
5 (-3, -32)
50 (, -23 (-17, 91))
41
91 (59 (75, -46), )
55 (, 62 (-31 (-10, 69), -74 (67, )))
-56
12 (96 (-22 (88, ), 31 (15, -92)), -47 (70, ))
-58 (4, -1 (27, -35))
78
-91 (89 (35 (-95, -24), -50 (, 77)), -95)
-69
89 (-93 (, -72), -31 (-76, -91))
-25 (93, 76)
32 (-71, 73 (-68 (, -12 (, -70)), -86 (-61 (-68, 58), 3-39))
68 (-10 (22, 60), 91)
89 (-7 (-20, 37), )
```

## Exemple de sortida 2

4  
4  
4  
4  
2  
2  
1  
2  
3  
1  
4  
3  
1  
4  
3  
2  
5  
3-39))  
2

## Observació

La vostra funció i subfuncions que creu 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 : 2023-10-27 14:21:09

© Jutge.org, 2006–2023.

<https://jutge.org>