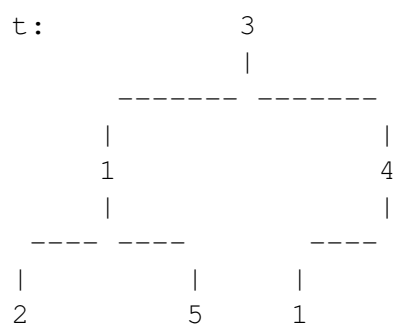

Suma dels valors d'un arbre**X26532_ca**

Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna la suma dels seus valors. Aquesta és la capcelera:

```
// Pre:  
// Post: Retorna la suma dels valors de t  
int sumOfTree(BinTree<int> t);
```

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



=>

16

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cc`, `BinTree.hh`, `sumOfTree.hh`. Us falta crear el fitxer `sumOfTree.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Quan pugeu la vostra solució al jutge, només cal que pugeu un tar construït així:

```
tar cf solution.tar sumOfTree.cc
```

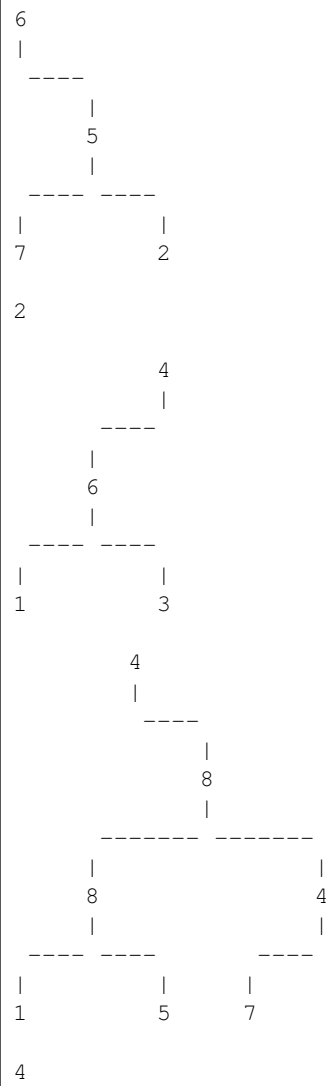
Entrada

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 la corresponent suma de l'arbre. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta suma. Només cal que implementeu la funció abans esmentada.

VISUALFORMAT



Exemple de sortida 1

27
46
47
28

14
20
2
14
37
4

Exemple d'entrada 2

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

Exemple de sortida 2

27
46
47
28
14
20
2
14
37
4

Observació

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

Informació del problema

Autoria: PRO2

Generació: 2026-01-25T14:44:05.605Z

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