

Arbre de sumes

X13275_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 conté la suma de nodes del subarbre que penja d'aquella mateixa posició a l'arbre inicial. Aquesta és la capcelera:

```
// Pre:
// Post: Retorna un arbre d'enters amb la mateixa estructura que t,
//       i a on cada subarbre té com a arrel la suma dels nodes del corresponen
BinTree<int> treeOfSums(BinTree<int> t);
```

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

```
treeOfSums(          3          ) =>          22
                   |
          -----
         |         |         |         |
        1         3         6         13
         |         |         |         |
        -----
       |         |         |         |
      5         2         5         10
       |         |         |         |
      -----
     |         |         |         |
    1         7         1         7
```

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

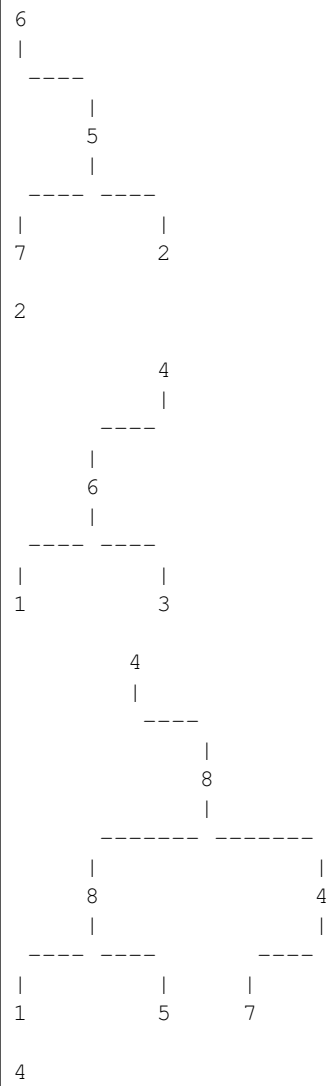
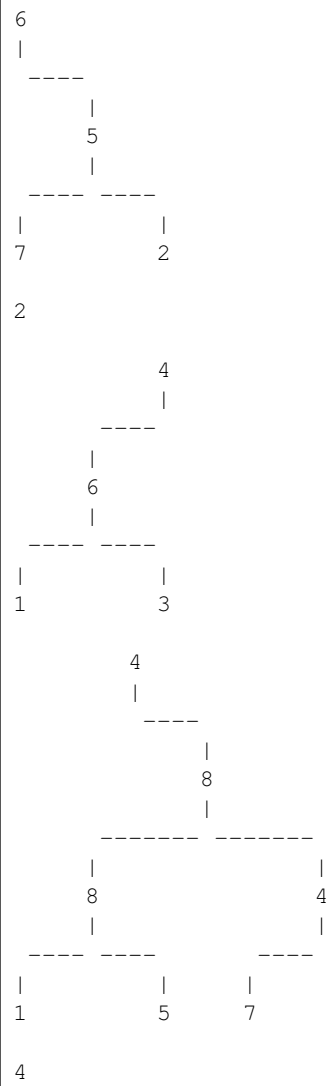
Entrada

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 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.

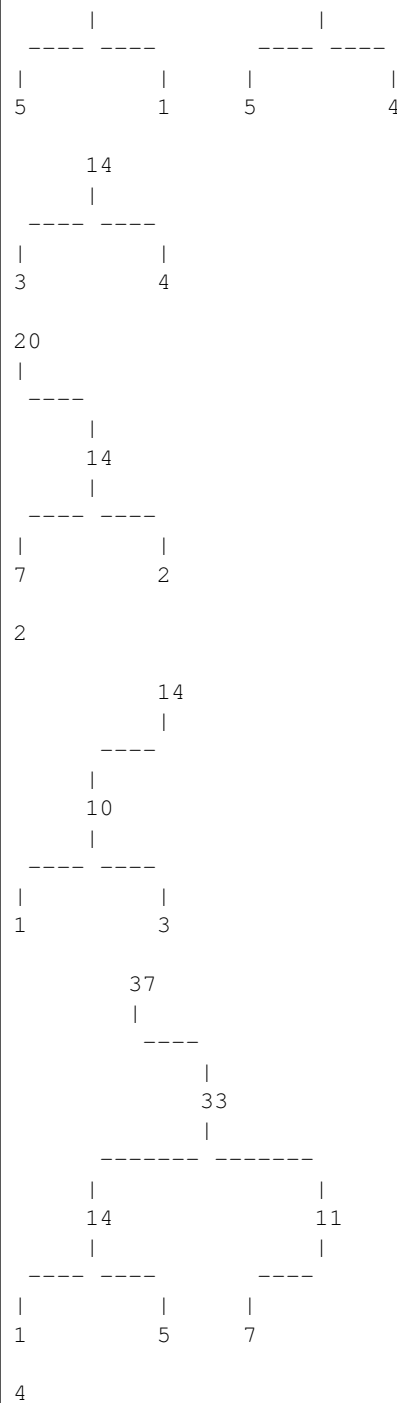
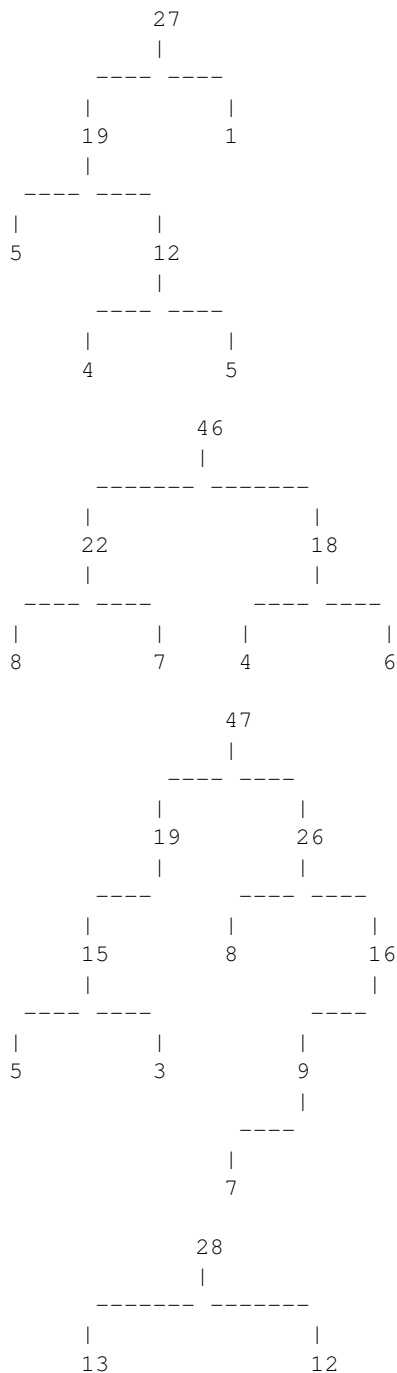
Sortida

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.

VISUALFORMAT



Exemple de sortida 1



Exemple d'entrada 2

INLINEFORMAT

0 (55 (29 (-47 (-15, 98),), -18 (86 (-59 (60 (29 (, -38 (34 (53 (87 (96 (-16 (-35, 37 (21 (12 (-28 (20) 52 (56, 58), -79) 75 (-46 (-53 (-48, -53), 98 (, 61),), -49) 67 (25, -50) 9 (-87, 25 (95,)) 15 (-92 (-47 (70,), -87),) 4 (-1 (27, -35),) 78 (86 (-5 (, 68),), 46 (88 (-59, -9 (68, 83)), 79 (89 (-53 (19, 33 (9 (24 (-75, 87), 180 (21 (-7, -16),))), 62 (-37 (90 (47, 28 (-25 (93 (76 (4, -8), -51 (-22 (-3, 21), 31 (-34, 32)))))

94 (37 (, 6), 72 (-90 (, 24 (, -38 (55 (-65, 22), 46))) , 38 (69 (22 (-65, 58 -20 (82, 81 (-19, 37))) 37 (34 (53 (87 (96 (-16 (-35, 37 (21 (12 (-28 (20) 52 (56, 58), -79) 62 (-37 (90 (47, 28 (-25 (93 (76 (4, -8), -51 (-22 (-3, 21), 31 (-34, 32))))) -6 (-10 (, 25 (80, 6 (57, 47))) , -60 (80, 87)) 40 (-71 (4 (-17 (90 (, -4 (, -57)) , -67 (, -87)) , 100), 20 (14 (-28, 80 (-14 (-95 (-31 (41 (-30 (59 (-71 (27, -4), -75 (, -92)) ,) , 59), -42) , 8 (54 (11 (-99 (67 (7,),), -47 (-10, -18)) , 82 (9, -9)) , 43 (16, -56) , -69 (-15 (25 (57 (38 (-54, -13), 80), -5), 39 (, -5 (-28 (-34,), 74 (, -53 (19, 33 (9 (24 (-75, 87), 180 (21 (-7, -16),))), 62 (-37 (90 (47, 28 (-25 (93 (76 (4, -8), -51 (-22 (-3, 21), 31 (-34, 32))))) , -95 (-40 (, 53), 93 (, -81 (16 (-61, 13 (89,)) , -7 (-20, 37))))))

