

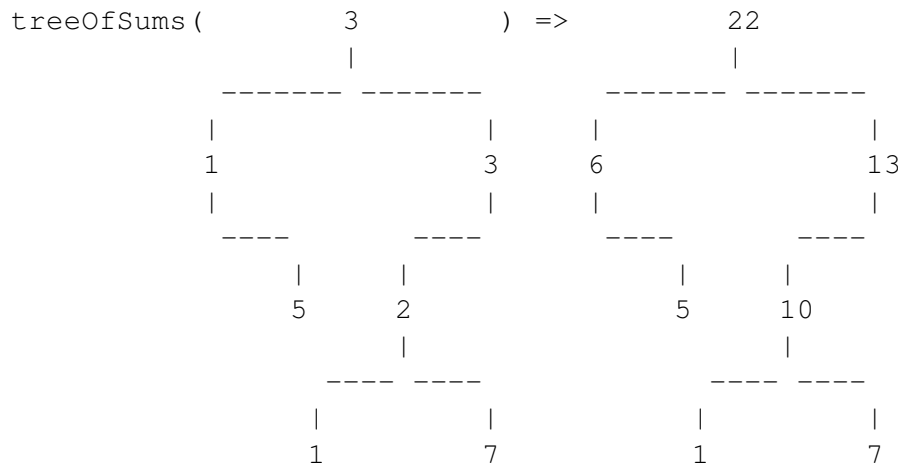
Arbre de sumes

X63359_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 capçalera:

```
// 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:



Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cc`, `BinTree.hh`, `treeOfSums.hh`. Us falta crear el fitxer `treeOfSums.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 treeOfSums.cc
```

Entrada

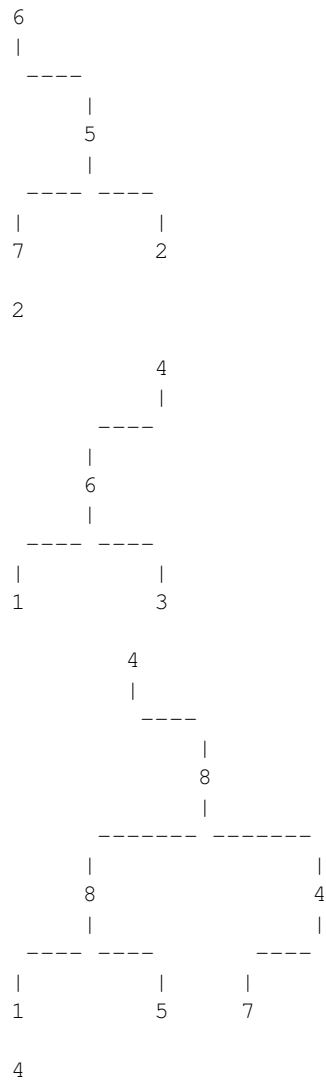
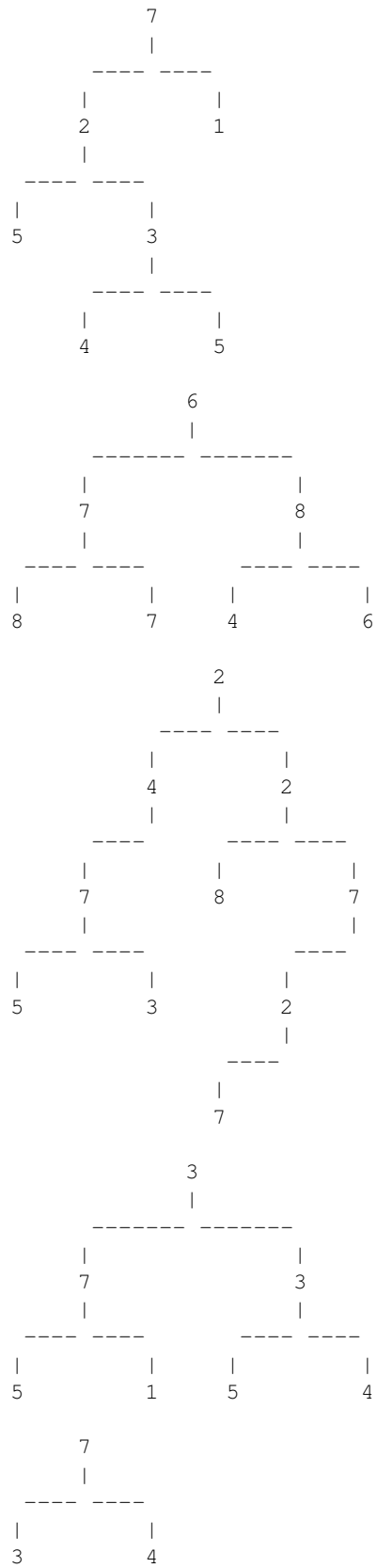
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `IN-LINEFORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega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

Exemple d'entrada 1

VISUALFORMAT



Exemple de sortida 1

```

      27
      |
    -----
    |  |
    19  1
    |
  -----
  |  |
  5   12
  |
    -----
    |  |
    4   5
    |
      46
      |
    -----
    |  |
    22  18
    |  |
  -----
  |  |  |  |
  8   7  4   6
  |
      47
      |
    -----
    |  |
    19  26
    |  |
  -----
  |  |  |
  15  8  16
  |  |  |
  -----
  |  |  |
  5   3  9
  |  |  |
  -----
  |  |
  |  7
  |
      28
      |
    -----
    |  |
    13  12
  
```

Exemple d'entrada 2

```

INLINEFORMAT
0 (55 (29 (-47 (-15, 98),), -18 (86 (-59 (60 (29 (,
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 (8
-25 (93 (76 (4, -8), -51 (-22 (-3, 21), 31 (-34, 32))

```

```

      |
    -----
    |  |
    5   1   5   4
    |
      14
      |
    -----
    |  |
    3   4
    |
      20
      |
    -----
    |  |
    |  14
    |
  -----
  |  |
  7   2
  |
      2
      |
        14
        |
        -----
        |  |
        10
        |
        -----
        |  |
        1   3
        |
          37
          |
          -----
          |  |
          |  33
          |
          -----
          |  |
          14  11
          |  |
        -----
        |  |  |
        1   5   7
        |
          4
        
```

```

94 (37 (, 6), 72 (-90 (, 24 (, -38 (55 (-65, 22), 46))), 38 (69 (22 (-6
58
-20 (82, 81 (-19, 37)))
38 (, 34 (53 (87 (80 (29 (, 62 (-21 (2 (12 (-28 (24), 32 (56 (, 58), 79)
-6 (-10 (, 25 (80, 6 (57, 47))), -60 (80, 87))
40 (-71 (4 (-17 (90 (, -4 (, -57))), -67 (, -87))), 100), 20 (14 (-28, 8
-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 (29 (75, 89))), 100 (21 (-7, -16),),), 62 (-37 (90 (47, 2
), -95 (-40 (, 53), 93 (, -81 (16 (-61, 13 (89,)), -7 (-20, 37))))))

```

40 (-49 (-36, -47 (51 (-22 (-7 (-67 (74 (33, -100), 15 (3, 53 (5, -65),),), 74 (-100, -88)), 42 (, 3
-9 (-64 (16,), 49 (-79, 74))

Exemple de sortida 2

-271 (-263 (65 (36 (-15, 98),), -383 (-211 (-100 (81 (-9 (, -38), 3
-15 (-41 (-154 (-48, -53), 159 (, 61)), -49)
42 (25, -50)
42 (-87, 120 (95,))
-141 (-156 (23 (70,), -87),)
-5 (-9 (27, -35),)
249 (149 (63 (, 68),), 22 (171 (-59, 142 (68, 83)), -195 (-76 (-93,
111 (139 (72 (4, -8), -26 (-4 (-3, 21), 29 (-34, 32))), -3 (13 (, 53)
830 (43 (, 6), 693 (-46 (, 44 (, 20 (12 (-65, 22), 46))), 667 (74 (-55
58
161 (82, 99 (-19, 37))
792 (576 (579 (526 (197 (23 (-35, 74 (, -23)), 270 (97, 108 (56,)))
306 (205 (, 215 (80, 110 (57, 47))), 107 (80, 87))
124 (-66 (-38 (-142 (29 (, -61 (, -57)), -154 (, -87)), 100), 43 (66
21 (-138 (-159 (-86 (-186 (-156 (-48 (27, -4), -167 (, -92))),), 59
58 (47 (-89 (-25 (74 (7,),), -75 (-10, -18)), 82 (9, -9)), 3 (16, -5
225 (129 (128 (108 (-29 (-54, -13), 80), -5), 16 (, -23 (-62 (-34,)
405 (19, 439 (58 (111 (-5, 87), -62 (-2 (-7, -16),),), 346 (224 (165
-627 (-536 (-36, -451 (-218 (-155 (-127 (-42 (7 (33, -100), 18), -
-13 (-48 (16,), 44 (-79, 74))

Observació

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

Informació del problema

Autor : PRO1

Generació : 2023-10-17 01:11:36

© *Jutge.org*, 2006–2023.

<https://jutge.org>