
Alçada d'un arbre

X16827_ca

Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna la seva alçada. L'alçada d'un arbre és el nombre de nodes que es troben en el camí més llarg des de l'arrel fins a alguna de les fulles. Noteu que, si l'arbre és buit, llavors té alçada 0, i si l'arbre té un únic node (que serà arrel i fulla alhora), llavors té alçada 1. Aquesta és la capçalera:

```
// Pre:  
// Post: Retorna l'alçada de t  
int heightOfTree(BinTree<int> t);
```

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

```
heightOfTree (      3      ) => 4  
                |  
            -----  
            |                |  
            1                3  
            |                |  
            -----  
                |          |  
                5          2  
                        |  
                        -----  
                            |  
                            1
```

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

Entrada

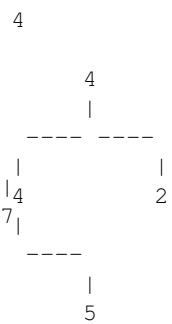
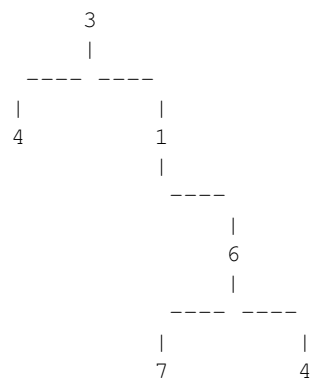
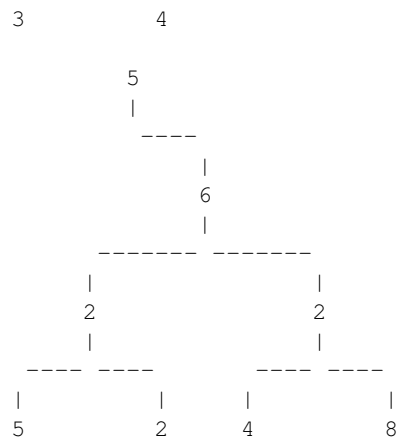
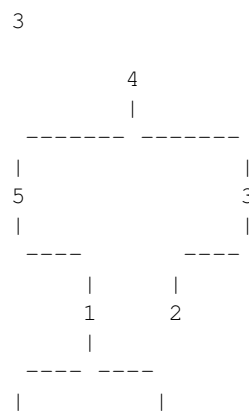
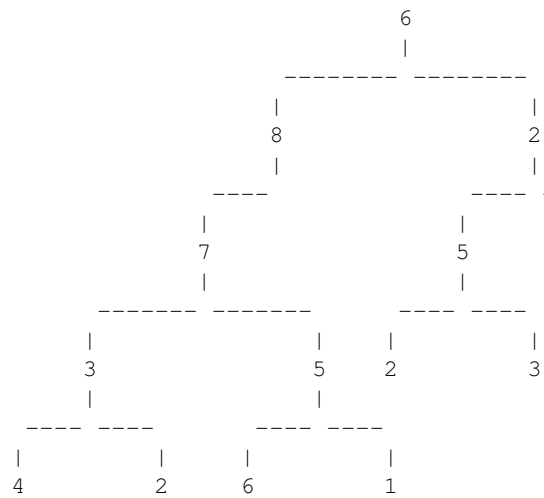
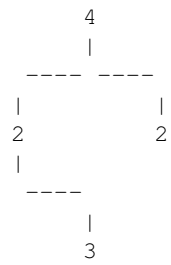
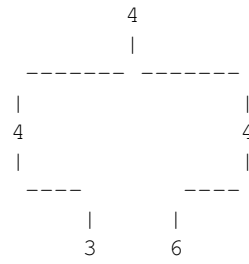
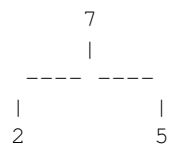
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `IN-LINEFORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir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

Sortida

Per a cada cas, la sortida conté la corresponent alçada de l'arbre. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta alçada. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

VISUALFORMAT



Exemple de sortida 1

2
3
3
5

1
4
4
4
1
3

Exemple d'entrada 2

INLINEFORMAT

```
0 (55 (29 (-47 (-15, 98) , ) , -18 (86 (-59 (60 (29 (, 34) , 30) , -13 (-80, -29) ) , 62 (-21, 2 (12 (-28, -20) , -67 (-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 (-93, -72) , -31 (-76, -91) ) ) )
-25 (93 (76 (4, -8) , -51 (-22 (-3, 21) , 31 (-34, 32) ) ) , -95 (-40 (, 53) , 93 (, -81 (16 (-61, 13 (89, ) ) , -7 (-20, 37) ) ) ) )
94 (37 (, 6) , 72 (-90 (, 24 (, -38 (55 (-65, 22) , 46) ) ) ) , 138 (69 (22 (-65, -12) , -54 (49 (78, -10) , -3) ) , 52 (56, 39 (80 (, 24)
58
-20 (82, 81 (-19, 37) )
97 (-45 (53 (87 (-96 (-16 (-35, 97 (, -23) ) , 65 (97, 52 (56, ) ) ) , 59 (20 (55 (77, -30) , ) , 61) ) , ) , -26 (98 (, 15) , 48 (, -71)
-6 (-10 (, 25 (80, 6 (57, 47) ) ) , -60 (80, 87) )
40 (-71 (4 (-17 (90 (, -4 (, -57) ) , -67 (, -87) ) ) , 100 (120 (14 (-28, 80) , -11 (-30, -2) ) ) , 70 (80, ) )
-14 (-95 (-31 (41 (-30 (59 (-71 (27, -4) , -75 (, -92) ) ) , 59) , -42) , 13 (31 (, -79) , -24 (62 (52 (80, -94 (, -60) ) , 26 (, 3
8 (54 (11 (-99 (67 (7, ) ) , -47 (-10, -18) ) ) , 82 (9, -96) , 43 (16, -56) )
-69 (-15 (25 (57 (38 (-54, -13) , 80) , -5) , 39 (, -5 (28 (-34, ) , 74 (-30, ) ) ) ) , 67 (41 (4, ) , -19 (72, ) ) )
-53 (19, 35 (9 (29 (-5, 87) , -60 (21 (-7, -16) , ) ) , 621 (-37 (90 (47, 28) , -35 (91, 40) , 60) ) )
40 (-49 (-36, -47 (51 (-22 (-7 (-67 (74 (33, -100) , 13) ) , -91 (13, ) ) , -69 (73 (-3, 53 (5, -65) , ) ) , 74 (-100, -88) ) , 42 (
-9 (-64 (16, ) , 49 (-79, 74) )
```

Exemple de sortida 2

8
2
3
4
3
5
97 (-93, -72) , -31 (-76, -91)))
9 , -95 (-40 (, 53) , 93 (, -81 (16 (-61, 13 (89,)) , -7 (-20, 37)))))
138 (69 (22 (-65, -12) , -54 (49 (78, -10) , -3)) , 52 (56, 39 (80 (, 24)
3
8
7
120 (14 (-28, 80) , -11 (-30, -2))) , 70 (80,))
6 ,) , 59) , -42) , 13 (31 (, -79) , -24 (62 (52 (80, -94 (, -60)) , 26 (, 3
96) , 43 (16, -56))
28 (-34,) , 74 (-30,)))) , 67 (41 (4,) , -19 (72,)))
1 (-37 (90 (47, 28) , -35 (91, 40) , 60)))
13) , -91 (13,)) , -69 (73 (-3, 53 (5, -65) ,)) , 74 (-100, -88)) , 42 (

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 : 2023-10-21 13:45:10

© [Jutge.org](https://jutge.org), 2006–2023.

<https://jutge.org>