

---

## Suma dels valors d'un arbre a profunditat parell

X21451\_ca

---

Implementeu una funció **RECURSIVA** que, donat un arbre binari d'enters, retorna la suma dels seus valors que es troben a profunditat parell. L'arrel es troba a profunditat 0 (parell), els nodes dels seus fills a profunditat 1 (no parell), i així successivament. Aquesta és la capçalera:

```
// Pre:  
// Post: Retorna la suma dels valors de t a profunditat parell  
int sumAtDepthEven(BinaryTree<int> t);
```

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

```
3(1(,5),4(1,)) => 9
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `sumAtDepthEven.hpp`. Us falta crear el fitxer `sumAtDepthEven.cpp` 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 sumAtDepthEven.cpp
```

### Entrada

L'entrada té un nombre arbitrari de casos. Cada cas consisteix en una línia amb un string describint 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 dels nodes de l'arbre a profunditat parell. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta suma. Només cal que implementeu la funció abans esmentada.

#### Exemple d'entrada 1

```
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 1

```
15  
31  
31  
18  
7  
15  
2  
8  
16  
4
```

## Exemple d'entrada 2

```
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) , -200) ) )
68 (-10 (22, 60) , 91)
89 (-7 (-20, 37) , )
```

## Exemple de sortida 2

```
-18
-57
-20
-281
5
124
41
120
-50
-56
91
-66
78
-106
-69
-150
-25
150
106
```

## Observació

La vostra funció i subfuncions que creu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema. En les crides recursives, incloeu la hipòtesi d'inducció, és a dir una explicació del que es compleix després de la crida, i també la funció de fita/decreixement o una justificació de perquè la funció recursiva acaba.

Necessitareu crear alguna funció recursiva auxiliar amb més paràmetres per tal d'abordar més fàcilment el problema i també per a produir una solució més eficient capaç de superar tots els jocs de proves.

## Informació del problema

Autor : PRO1

Generació : 2022-03-21 20:02:08

© *Jutge.org*, 2006–2022.

<https://jutge.org>