

## Arbre de mides

X61092\_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 el nombre total 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 valor a l'arrel el nombre de nodes  
//        del corresponent subarbre a t.  
BinaryTree<int> treeOfSizes(BinaryTree<int> t);
```

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

```
treeOfSums(3(1(,5),3(2(1,7),))) => 7(2(,1),4(3(1,1),))
```

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

```
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

```
7(5(1,3(1,1)),1)  
7(3(1,1),3(1,1))  
10(4(3(1,1),),5(1,3(2(1,),)))  
7(3(1,1),3(1,1))  
3(1,1)  
4(,3(1,1))  
1  
4(3(1,1),)  
7(,6(3(1,1),2(1,)))  
1
```

## Exemple d'entrada 2

```
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(891(-7342721),,-31137541,9311,1)),,7(3(1,1),3(1,1))),-25(93(76(4,-8),,-51(-22(-3,21),31(-34,32))24(-195(3-41,(1),531(-3)81,,1813(16,(1-6)1),1B2(821,1),,-9((r-20481),211,94(37(,6),72(-90(,24(,-38(55(-65,22),46))3782(6,91(2,23(4-65,-42),5(-541(A9)(7B),,-1026+2134521(56589((B01),2158-20(82,81(-19,37))5(1,3(1,1))97(-45(53(87(-96(-16(-35,97(,-23)),,65(97,-6(-10(,25(80,6(57,47))),,-60(80,87))40(-71(4(-17(90(,-4(,-57)),,-67(,-87)),,100)1201(6,08(+283,8,02,-111)(-20,1-2),,1),7D((301,1),,3(1,1)),,2(1,)),-14(-95(-31(41(-30(59(-71(27,-4),,-75(,-92)72(25,01,1+12)(6,83(81,(1)-721,1-2)4,(621(521(8017-02((,1-6014(264(88(54(11(-99(67(7,),,-47(-10,-18)),,82(9,-915(413(7,63(-56),),,3(1,1)),,3(1,1)),,3(1,1)),-69(-15(25(57(38(-54,-13),80),,-5),,39(,-5(-228((1-47,15734((1,-3D),,1)),,14,676(415(4211,-1,92(72,))),,5(2(1,),,2(1,-53(19,35(9(29(-5,87),,-60(21(-7,-16),)),,622(0-81,(180((17328),,-35((311AD))),,6,09(7(3(1,1),3(1,1)),,1)))40(-49(-36,-47(51(-22(-7(-67(74(33,-100),,11892(-511((1,349(19-695(73((5-6353,(5,-165),211),,7A8(+50D,38B),1),42(-9(-64(16,),,49(-79,74))
```

## Exemple de sortida 2

```
-3E3,4B014(-3,81,-B0,,)-29611628(-212,0,(112,(1283+201),,-69((1-58,379))8(6(3(1,1),2(,1)),,1)3(1,1)4(1,2(1,))5(4(2(1,),,1),,)4(3(1,1),)15(1,3(1,1))10(6(,5(1,3(1,1))),,3(1,1))1201(6,08((204(55,(27,1)30),4(1,6211,1)),,26((983(151),A8,(1)-71,526(6264017,158((204(55,(27,1)30),4(1,6211,1)),,26((983(151),A8,(1)-71,-6(-10(,25(80,6(57,47))),,-60(80,87))10(6(,5(1,3(1,1))),,3(1,1))1201(6,08(+283,8,02,-111)(-20,1-2),,1),7D((301,1),,3(1,1)),,2(1,)),-14(-95(-31(41(-30(59(-71(27,-4),,-75(,-92)72(25,01,1+12)(6,83(81,(1)-721,1-2)4,(621(521(8017-02((,1-6014(264(88(54(11(-99(67(7,),,-47(-10,-18)),,82(9,-915(413(7,63(-56),),,3(1,1)),,3(1,1)),,3(1,1)),-69(-15(25(57(38(-54,-13),80),,-5),,39(,-5(-228((1-47,15734((1,-3D),,1)),,14,676(415(4211,-1,92(72,))),,5(2(1,),,2(1,-53(19,35(9(29(-5,87),,-60(21(-7,-16),)),,622(0-81,(180((17328),,-35((311AD))),,6,09(7(3(1,1),3(1,1)),,1)))40(-49(-36,-47(51(-22(-7(-67(74(33,-100),,11892(-511((1,349(19-695(73((5-6353,(5,-165),211),,7A8(+50D,38B),1),42(-9(-64(16,),,49(-79,74))6(2(1,),,3(1,1))
```

## Observació

La vostra funció i subfuncions que creeu 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.

Molt possiblement, una solució directa serà lenta, i necessitareu crear alguna funció recursiva auxiliar per a produir una solució més eficient capaç de superar tots els jocs de proves.

## Informació del problema

Autor : PRO1

Generació : 2022-09-14 09:14:49

© Jutge.org, 2006–2022.

<https://jutge.org>