
Arbre de sumes

X22019_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 capçelera:

```
// 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  
BinaryTree<int> treeOfSums(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),))) => 22(6(,5),13(10(1,7),))
```

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

```
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

```
27(19(5,12(4,5)),1)  
46(22(8,7),18(4,6))  
47(19(15(5,3),),26(8,16(9(7,)),))  
28(13(5,1),12(5,4))  
14(3,4)  
20(,14(7,2))  
2  
14(10(1,3),)  
37(,33(14(1,5),11(7,)))  
4
```

Exemple d'entrada 2

```
0(55(29(-47(-15,98),),-18(86(-59(60(29(,38),39),-47(70,,-28))),),62(-21,2(12(-28,-20),-67(-58,-79))  
75(-46(-53(-48,-53),98(,61)),,-49)  
4(-1(27,-35),)
```

	Exemple de sortida 2
78 (86 (-5 (, 68) ,) , 46 (88 (-59, -9 (68, 83)) , 79 (89 (-93, -72) , -31 (-76, -91))))	27195 (-263 (65536 6315 (-88) (16 (-383 (13289 (110481 (-209 67738))))))
-25 (93 (76 (4, -8) , -51 (-22 (-3, 21) , 31 (-34, 32))) , 58 (74 (22 (-545, -48)))) , 54 (59 (181 (-107493)) , 52 (56, 39 (80 (, 24	42 (25, -50)
58	42 (-87, 120 (95,))
-20 (82, 81 (-19, 37))	-141 (-156 (23 (70 (55 (787, 130) ,) , 61)) ,) , -26 (98 (, 15) , 48 (, -71
97 (-45 (53 (87 (-96 (-16 (-35, 97 (, -23)) , 65 (97, 52 (56 (, 1)) , 59 (20 (65 (787, 130) ,) , 61)) ,) , -26 (98 (, 15) , 48 (, -71	-5 (-9 (27, -35) ,)
-6 (-10 (, 25 (80, 6 (57, 47))) , -60 (80, 87))	249 (149 (63 (868) , 112 (26171 (2759, 142 (668, 83)) , -195 (-76 (-93,
40 (-71 (4 (-17 (90 (, -4 (, -57)) , -67 (, -87)) , 100 (111 (139 (74 (24, 78) (3126 (-74) , -324 (622 (32780, 324)) , -68 (13 6533	8 (54 (11 (-99 (67 (7,) ,) , -47 (-10, -18)) , 82 (9, 98) 30 (43 (16 (-5693 (-46 (, 44 (, 20 (12 (-65, 22) , 46))) , 667 (74 (-55
8 (54 (11 (-99 (67 (7,) ,) , -47 (-10, -18)) , 82 (9, -58 (-34,) , 74 (-30,))) , 67 (41 (4,) , -19 (72,)))	-28 (-34,) , 74 (-30,))) , 67 (41 (4,) , -19 (72,)))
-69 (-15 (25 (57 (38 (-54, -13) , 80) , -5) , 39 (, -5	21 (63 (82099 (1803735 (91, 40)) , 60)))
-53 (19, 35 (9 (29 (-5, 87) , -60 (21 (-7, -16) ,)) , 6	1892 (5716 (579 (526 (1973 (23 (535674 65723))) , 270 (170108 859 (42)
40 (-49 (-36, -47 (51 (-22 (-7 (-67 (74 (33, -100)) , 18	306 (205 (, 215 (80, 110 (57, 47))) , 107 (80, 87))
-9 (-64 (16,) , 49 (-79, 74))	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. 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 08:52:23

© Jutge.org, 2006–2022.

<https://jutge.org>