Reemplaça els nodes d'un arbre binari a profunditat parell per la suma per sota X68048_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 a profunditat parell conté la suma de nodes del subarbre que penja d'aquella mateixa posició a l'arbre inicial, i a cada posició a profunditat senar hi ha exactament el mateix valor que es troba en aquella posició a l'arbre inicial.
Sobreentenem que l'arrel de l'arbre està a profunditat 0, els nodes directes des de l'arrel són a profunditat 1, els nodes a distància dos de l'arrel són a profunditat 2, i així successivament.
Aquesta és la capcelera:

```
/**
 * @brief Retorna l'arbre binari `t` reemplaçant els valors dels nodes
 * a profunditat parell per la suma per sota
 *
 * @param t L'arbre binari original.
 *
 * @returns Un arbre binari R amb la mateixa estructura que t.
 * Per a cada posició p de t i R, si p és a profunditat senar,
 * llavors t i R tenen el mateix valor a posició p.
 * En canvi, si p es a profunditat parell, llavors el valor de R a posició
 * p és la suma de tots els valors que es troben a t a posició p i per sota.
 */
BinTree<int> sum_below_at_even_depth(BinTree<int> t);
```

Observació

Els fitxers públics (icona del gatet) són: la classe `BinTree` (fitxer `bintree.hh`), l'entrada/sortida de `BinTree` (`bintree-io.hh` i `bintree-inline.hh`) i el programa principal. També hi ha un `Makefile` i el directori `.vscode` que té la configuració per compilar i debuggar amb VSCode.
Has d'implementar `sum_below_at_even_depth` en un **fitxer `.cc` nou**, compilar (està preparat per poder compilar i debuggar amb VSCode), i finalment **enviar només el fitxer amb la funció**.
La vostra funció i subfuncions que creeu han de treballar només amb arbres. Molt possiblement, una solució recursiva directa serà lenta, i necessitareu crear alguna funció recursiva auxiliar per a produïr una solució més eficient capaç de superar tots els jocs de proves.

Entrada

Cada cas consisteix en una representació textual d'un arbre binari d'enters. (Aquesta lectura ja la fa el programa principal.)

Sortida

Per a cada cas, la sortida conté la representació textual de l'arbre resultant d'aplicar la funció `sum_below_at_even_depth`. (La sortida també la fa el programa principal.)

Exemple d'entrada 1

```
visual
8
|-- 6
|   |-- 4
|   |   |-- 6
|   |   |   |-- 8
|   |   |   '-- 7
|   |   '-- #
|   '-- 2
|       |-- 7
|       |   |-- 9
|       |   '-- 2
|       '-- 1
|           |-- #
|           '-- 6
'-- 8
    |-- 7
    '-- 3

3
|-- 3
|   |-- 9
|   |   |-- 6
|   |   '-- #
|   '-- 5
'-- 6
    |-- #
    '-- 9
        |-- 5
        '-- #

1
|-- #
'-- 7
    |-- 7
    |   |-- 7
    |   '-- #
    '-- 6
        |-- 3
        |   |-- 7
        |   '-- #
        '-- #

2
|-- #
'-- 1
    |-- 1
    |   |-- #
    |   '-- 2
    '-- #

3
|-- 8
|   |-- 3
|   '-- #
'-- 7
    |-- 6
    '-- 1
        |-- 3
        |   |-- 1
        |   '-- 8
        '-- 7

6
|-- 4
|   |-- 4
|   |   |-- 1
|   |   |   |-- 5
|   |   |   '-- #
|   |   '-- 3
|   |       |-- 3
|   |       '-- #
|   '-- #
'-- 3
    |-- 9
    |   |-- 6
    |   |   |-- 9
    |   |   '-- 8
    |   '-- 7
    |       |-- 5
    |       '-- 5
    '-- 2
        |-- 4
        |   |-- 8
        |   '-- 4
        '-- 3
            |-- 7
            '-- 2

3
|-- 8
'-- 1
    |-- 3
    |   |-- #
    |   '-- 4
    |       |-- 9
    |       '-- #
    '-- 3

3
|-- 4
|   |-- 9
|   |   |-- 1
|   |   '-- 6
|   '-- 6
|       |-- #
|       '-- 6
|           |-- #
|           '-- 9
'-- 8
    |-- 8
    '-- 9
        |-- 7
        '-- 2
            |-- #
            '-- 6

1
|-- #
'-- 6
```

```
    |-- #
    '-- 3
        |-- 9
        '-- 8


8
|-- 5
|   |-- 2
|   |   |-- #
|   |   '-- 8
|   |       |-- #
|   |       '-- 3
|   '-- 3
|       |-- #
|       '-- 7
|           |-- 7
|           '-- 6
'-- 2
```

Exemple de sortida 1

```
84
|-- 6
|   |-- 25
|   |   |-- 6
|   |   |   |-- 8
|   |   |   '-- 7
|   |   '-- #
|   '-- 27
|       |-- 7
|       |   |-- 9
|       |   '-- 2
|       '-- 1
|           |-- #
|           '-- 6
'-- 8
    |-- 7
    '-- 3

46
|-- 3
|   |-- 15
|   |   |-- 6
|   |   '-- #
|   '-- 5
'-- 6
    |-- #
    '-- 14
        |-- 5
        '-- #

38
|-- #
'-- 7
    |-- 14
    |   |-- 7
    |   '-- #
    '-- 16
        |-- 3
        |   |-- 7
        |   '-- #
        '-- #

6
|-- #
'-- 1
    |-- 3
    |   |-- #
    |   '-- 2
    '-- #

47
|-- 8
|   |-- 3
|   '-- #
'-- 7
    |-- 6
    '-- 20
        |-- 3
        |   |-- 1
```

```
        |   '-- 8
        '-- 7


108
|-- 4
|   |-- 16
|   |   |-- 1
|   |   |   |-- 5
|   |   |   '-- #
|   |   '-- 3
|   |       |-- 3
|   |       '-- #
|   '-- #
'-- 3
    |-- 49
    |   |-- 6
    |   |   |-- 9
    |   |   '-- 8
    |   '-- 7
    |       |-- 5
    |       '-- 5
    '-- 30
        |-- 4
        |   |-- 8
        |   '-- 4
        '-- 3
            |-- 7
            '-- 2


31
|-- 8
'-- 1
    |-- 16
    |   |-- #
    |   '-- 4
    |       |-- 9
    |       '-- #
    '-- 3


84
```

```
|-- 4
|   |-- 16
|   |   |-- 1
|   |   '-- 6
|   '-- 21
|       |-- #
|       '-- 6
|           |-- #
|           '-- 9
'-- 8
    |-- 8
    '-- 24
        |-- 7
        '-- 2
            |-- #
            '-- 6


27
|-- #
'-- 6
    |-- #
    '-- 20
        |-- 9
        '-- 8


51
|-- 5
|   |-- 13
|   |   |-- #
|   |   '-- 8
|   |       |-- #
|   |       '-- 3
|   '-- 23
|       |-- #
|       '-- 7
|           |-- 7
|           '-- 6
'-- 2
```

Exemple d'entrada 2

```
inline
8(6(4(6(8,7),),2(7(9,2),1(,6))),8(7,3))
3(3(9(6,),5),6(,9(5,)))
1(,7(7(7,),6(3(7,),)))
2(,1(1(,2),))
3(8(3,),7(6,1(3(1,8),7)))
6(4(4(1(5,),3(3,)),),3(9(6(9,8),7(5,5)),2(31(8,4),1(,6(7,4)))),3))
3(8,1(3(,4(9,)),3))
3(4(9(1,6),6(,6(,9))),8(8,9(7,2(,6))))
1(,6(,3(9,8)))
8(5(2(,8(,3)),3(,7(7,6))),2)
```

Exemple de sortida 2

```
84(6(25(6(8,7),),27(7(9,2),1(,6))),8(7,3))
46(3(15(6,),5),6(,14(5,)))
38(,7(14(7,),16(3(7,),)))
6(,1(3(,2),))
47(8(3,),7(6,20(3(1,8),7)))
108(4(16(1(5,),3(3,)),),3(49(6(9,8),7(5,5)),30(4(8,4),
84(4(16(1,6),21(,6(,9))),8(8,24(7,2(,6))))
27(,6(,20(9,8)))
51(5(13(,8(,3)),23(,7(7,6))),2)
```

Informació del problema

Autor : PRO2
Generació : 2025-03-22 16:25:21