Suma i mida de molts arbres X68237_ca

En aquest exercici, heu d'implementar un programa que llegeix comandes que manipulen variables que guarden àrbres binaris d'enters. La primera comanda `numvars=` *n* `;` indica el nombre total *n* de variables. Els noms d'aquestes variables son `t0,...,t(n-1)`, i se suposa que inicialment cadascuna guarda un àrbre buit. Després venen comandes que construeixen nous àrbres a partir de variables i els assignen a variables (com per exemple `t2 =BinTree( 3 , t0 , t1 );`, i comandes que accedeixen als fills d'un arbre existent i els assignen a variables (com per exemple `t3 = t2 .left();` o `t3 = t2 .right();`). També hi ha comandes per a escriure per la sortida un àrbre en `INLINEFORMAT` (com per exemple `cout<< t2 <<endl;`), i instruccions per a escriure la mida o la suma dels valors d'un arbre guardat en una variable, com per exemple (`cout<<size( t2 )<<endl;` o `cout<<sum( t2 )<<endl;`).
Aquest és un exemple d'entrada del programa:

```
numvars= 4 ;
t1 =BinTree( 1 , t2 , t3 );
t2 =BinTree( 2 , t1 , t3 );
t3 =BinTree( 3 , t2 , t1 );
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<<size( t0 )<<endl;
cout<<size( t1 )<<endl;
cout<<size( t2 )<<endl;
cout<<size( t3 )<<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t1 )<<endl;
cout<<sum( t2 )<<endl;
cout<<sum( t3 )<<endl;
t1 =BinTree( 1 , t2 , t3 );
t2 =BinTree( 2 , t1 , t3 );
t3 =BinTree( 3 , t2 , t1 );
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<<size( t0 )<<endl;
cout<<size( t1 )<<endl;
cout<<size( t2 )<<endl;
cout<<size( t3 )<<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t1 )<<endl;
cout<<sum( t2 )<<endl;
cout<<sum( t3 )<<endl;
t1 = t3 .left();
```

```
t2 = t1 .right();
t3 = t2 .left();
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<<size( t0 )<<endl;
cout<<size( t1 )<<endl;
cout<<size( t2 )<<endl;
cout<<size( t3 )<<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t1 )<<endl;
cout<<sum( t2 )<<endl;
cout<<sum( t3 )<<endl;
```

La sortida del programa amb la seqüència de comandes d'entrada anterior hauria de ser:

```
()
1
2(1,)
3(2(1,),1)
0
1
2
4
0
1
3
7
()
1(2(1,),3(2(1,),1))
2(1(2(1,),3(2(1,),1)),3(2(1,),1))
3(2(1(2(1,),3(2(1,),1)),3(2(1,),1)),1(2(1,),3(2(1,),1)))
0
7
12
20
0
11
20
34
()
2(1(2(1,),3(2(1,),1)),3(2(1,),1))
3(2(1,),1)
2(1,)
0
12
4
2
0
```

```
20
7
3
```

Com podeu observar a l'exemple d'entrada anterior, hi han espais en blanc per a facilitar la lectura. Podeu llegir i tractar les comandes així:

```
#include <iostream>
#include <string>
#include <cstdlib>
//...

using namespace std;

#include "BinTree.hh"

int getIdVar(string s)
{
return atoi(s.substr(1).c_str());
}

//...

int main()
{
//...
string s1, s2, s3, s4, s5, s6, s7;
int numvars;
cin >> s1 >> numvars >> s2;
// ...
while (cin >> s1 >> s2) {
if (s1[0] == 't') {
int idvar = getIdVar(s1);
if (s2 == "=BinTree(") {
int value;
cin >> value >> s3 >> s4 >> s5 >> s6 >> s7;
int idvar1 = getIdVar(s4);
int idvar2 = getIdVar(s6);
//...
} else if (s2 == "=") {
cin >> s3 >> s4;
int idvar1 = getIdVar(s3);
if (s4 == ".left();") {
//...
} else {
//...
}
}
} else if (s1 == "cout<<") {
int idvar = getIdVar(s2);
cin >> s3;
//...
//....setOutputFormat(BinTree<int>::INLINEFORMAT);
//cout << ... << endl;
} else if (s1 == "cout<<size(") {
int idvar = getIdVar(s2);
cin >> s3;
//...
} else if (s1 == "cout<<sum(") {
int idvar = getIdVar(s2);
cin >> s3;
//...
}
}
}
```

Fixeu-vos que l'enunciat d'aquest exercici us ofereix el fitxer `BinTree.hh`. Us falta crear el fitxer `main.cc`, que haurieu de construïr a partir de la plantilla que us hem oferit abans, fent un ús convenient del tipus `BinTree`. Només cal que pugeu `main.cc` al jutge.
Observació: Us recomanem que comenceu implementant una solució bàsica per tal de superar els jocs de proves públics i obtenir així la meitat de la nota. Ja la optimitzareu més endavant si teniu temps.

Entrada

La primera linia de l'entrada és de la forma `numvars= LIMIT ;`, a on LIMIT és un nombre natural positiu. Després venen instruccions d'aquestes menes:

```
tNUM =BinTree( VALUE , tNUM1 , tNUM2 );
tNUM1 = tNUM2 .left();
tNUM1 = tNUM2 .right();
cout<< tNUM <<endl;
cout<<size( tNUM )<<endl;
cout<<sum( tNUM )<<endl;
```

On `VALUE` es un enter i `NUM, NUM1, NUM2` son naturals en el rang {0,...,LIMIT-1}.
Se suposa que les entrades son correctes: sempre es demana accedir a `left` o `right` d'arbres no buits, i no es produeixen errors d'overflow.

Sortida

Per a cada instrucció dels següents tres tipus, el vostre programa ha d'escriure el resultat esperat (l'arbre contingut en la variable en `INLINEFORMAT`, o la mida de l'arbre contingut en la variable, o la suma de l'arbre contingut en la variable, segons el cas).

```
cout<< tNUM <<endl;
cout<<size( tNUM )<<endl;
cout<<sum( tNUM )<<endl;
```

Exemple d’entrada 1

```
numvars= 4 ;
t1 =BinTree( 1 , t2 , t3 );
t2 =BinTree( 2 , t1 , t3 );
t3 =BinTree( 3 , t2 , t1 );
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<<size( t0 )<<endl;
cout<<size( t1 )<<endl;
cout<<size( t2 )<<endl;
cout<<size( t3 )<<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t1 )<<endl;
cout<<sum( t2 )<<endl;
cout<<sum( t3 )<<endl;
t1 =BinTree( 1 , t2 , t3 );
t2 =BinTree( 2 , t1 , t3 );
t3 =BinTree( 3 , t2 , t1 );
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<<size( t0 )<<endl;
cout<<size( t1 )<<endl;
cout<<size( t2 )<<endl;
cout<<size( t3 )<<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t1 )<<endl;
cout<<sum( t2 )<<endl;
cout<<sum( t3 )<<endl;
t1 = t3 .left();
t2 = t1 .right();
t3 = t2 .left();
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<<size( t0 )<<endl;
cout<<size( t1 )<<endl;
cout<<size( t2 )<<endl;
cout<<size( t3 )<<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t1 )<<endl;
cout<<sum( t2 )<<endl;
cout<<sum( t3 )<<endl;
```

Exemple de sortida 1

```
()
1
2(1,)
3(2(1,),1)
0
1
2
4
0
1
3
7
()
1(2(1,),3(2(1,),1))
2(1(2(1,),3(2(1,),1)),3(2(1,),1))
3(2(1(2(1,),3(2(1,),1)),3(2(1,),1)),1(2(1,),3(2(1,),1)
0
7
12
20
0
11
20
34
()
2(1(2(1,),3(2(1,),1)),3(2(1,),1))
3(2(1,),1)
2(1,)
0
12
4
2
0
20
7
3
```

Exemple d’entrada 2

```
numvars= 3 ;
cout<< t1 <<endl;
cout<< t1 <<endl;
t1 =BinTree( 1 , t0 , t0 );
t1 =BinTree( 2 , t2 , t1 );
cout<<size( t1 )<<endl;
t0 = t1 .left();
t2 =BinTree( 5 , t0 , t0 );
t2 =BinTree( 2 , t1 , t1 );
t0 = t2 .right();
t1 = t1 .left();
t0 =BinTree( 4 , t0 , t0 );
t1 =BinTree( 2 , t2 , t2 );
cout<<size( t0 )<<endl;
cout<<sum( t1 )<<endl;
t2 =BinTree( 2 , t0 , t2 );
cout<< t1 <<endl;
cout<<size( t0 )<<endl;
t0 = t1 .right();
t0 =BinTree( 0 , t0 , t0 );
t1 = t0 .left();
t2 = t1 .right();
```

```
cout<< t0 <<endl;
t0 = t1 .left();
t0 =BinTree( 4 , t0 , t2 );
cout<<sum( t1 )<<endl;
cout<<sum( t1 )<<endl;
cout<< t0 <<endl;
t0 =BinTree( 2 , t0 , t1 );
t2 =BinTree( 1 , t2 , t1 );
cout<< t2 <<endl;
cout<<sum( t2 )<<endl;
t1 = t2 .right();
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
t2 = t0 .right();
cout<< t1 <<endl;
cout<<size( t1 )<<endl;
cout<<sum( t0 )<<endl;
cout<<size( t2 )<<endl;
t1 = t2 .right();
cout<<sum( t1 )<<endl;
cout<<size( t1 )<<endl;
t2 = t1 .left();
cout<< t1 <<endl;
t1 =BinTree( 3 , t1 , t2 );
cout<<sum( t2 )<<endl;
t1 =BinTree( 2 , t1 , t1 );
cout<< t2 <<endl;
cout<<sum( t0 )<<endl;
cout<<size( t2 )<<endl;
t1 =BinTree( 5 , t2 , t1 );
cout<< t2 <<endl;
t2 = t1 .right();
cout<<sum( t0 )<<endl;
t2 = t1 .left();
t2 =BinTree( 1 , t2 , t1 );
cout<< t2 <<endl;
cout<< t2 <<endl;
cout<<size( t1 )<<endl;
cout<<sum( t1 )<<endl;
cout<< t1 <<endl;
t1 = t2 .left();
t1 = t0 .right();
cout<<sum( t1 )<<endl;
t2 = t1 .right();
cout<< t1 <<endl;
t1 =BinTree( 2 , t1 , t2 );
cout<< t2 <<endl;
t2 = t0 .right();
t2 = t0 .right();
cout<<size( t1 )<<endl;
cout<< t1 <<endl;
cout<< t1 <<endl;
cout<< t1 <<endl;
cout<<sum( t1 )<<endl;
t1 = t1 .left();
cout<<size( t2 )<<endl;
cout<< t1 <<endl;
cout<< t0 <<endl;
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
```

Exemple de sortida 2

```
()
()
2
5
18
2(2(2(,1),2(,1)),2(2(,1),2(,1)))
5
0(2(2(,1),2(,1)),2(2(,1),2(,1)))
8
8
4(2(,1),2(,1))
1(2(,1),2(2(,1),2(,1)))
12
2(4(2(,1),2(,1)),2(2(,1),2(,1)))
2(2(,1),2(,1))
1(2(,1),2(2(,1),2(,1)))
2(2(,1),2(,1))
5
20
5
3
2
2(,1)
0
()
20
0
()
20
1(,5(,2(3(2(,1),),3(2(,1),))))
1(,5(,2(3(2(,1),),3(2(,1),))))
8
19
5(,2(3(2(,1),),3(2(,1),)))
8
2(2(,1),2(,1))
2(,1)
8
2(2(2(,1),2(,1)),2(,1))
2(2(2(,1),2(,1)),2(,1))
2(2(2(,1),2(,1)),2(,1))
13
5
2(2(,1),2(,1))
2(4(2(,1),2(,1)),2(2(,1),2(,1)))
2(4(2(,1),2(,1)),2(2(,1),2(,1)))
2(2(,1),2(,1))
2(2(,1),2(,1))
```

Exemple d'entrada 3

```
numvars= 10 ;
cout<< t6 <<endl;
cout<< t5 <<endl;
t6 =BinTree( -1 , t2 , t9 );
t7 =BinTree( 7 , t0 , t9 );
cout<<size( t6 )<<endl;
t8 =BinTree( 6 , t7 , t9 );
t2 =BinTree( -15 , t3 , t7 );
t9 = t2 .right();
t8 = t9 .right();
t3 = t6 .left();
t9 =BinTree( -1 , t3 , t1 );
t7 =BinTree( 4 , t8 , t4 );
cout<<size( t0 )<<endl;
cout<<sum( t6 )<<endl;
t6 =BinTree( 13 , t3 , t2 );
cout<< t6 <<endl;
cout<<size( t5 )<<endl;
t7 =BinTree( 6 , t6 , t5 );
t5 =BinTree( 9 , t2 , t5 );
t4 =BinTree( -2 , t4 , t3 );
t8 =BinTree( 6 , t4 , t3 );
t4 = t9 .right();
cout<< t0 <<endl;
t8 = t9 .left();
t6 =BinTree( -10 , t4 , t9 );
cout<<sum( t0 )<<endl;
cout<<sum( t8 )<<endl;
cout<< t1 <<endl;
t7 =BinTree( 18 , t2 , t2 );
t0 =BinTree( -7 , t6 , t1 );
cout<< t9 <<endl;
cout<<sum( t9 )<<endl;
t1 =BinTree( 20 , t7 , t7 );
t5 =BinTree( 0 , t9 , t7 );
t6 = t7 .right();
cout<< t6 <<endl;
cout<< t6 <<endl;
t4 =BinTree( 6 , t8 , t1 );
cout<< t9 <<endl;
t9 = t0 .right();
cout<< t8 <<endl;
cout<<size( t0 )<<endl;
cout<<sum( t6 )<<endl;
t5 =BinTree( 18 , t6 , t1 );
cout<<size( t5 )<<endl;
t8 = t4 .right();
cout<<sum( t1 )<<endl;
cout<<size( t3 )<<endl;
t4 = t4 .left();
cout<< t4 <<endl;
t3 =BinTree( 5 , t1 , t7 );
t6 =BinTree( 8 , t2 , t1 );
t5 =BinTree( -11 , t7 , t4 );
cout<<sum( t8 )<<endl;
t7 =BinTree( 19 , t5 , t3 );
t3 =BinTree( 12 , t1 , t8 );
t4 =BinTree( 19 , t3 , t3 );
cout<< t8 <<endl;
t4 =BinTree( -9 , t8 , t8 );
t7 =BinTree( 2 , t7 , t6 );
cout<<sum( t3 )<<endl;
cout<<size( t3 )<<endl;
t2 =BinTree( -9 , t5 , t4 );
cout<< t5 <<endl;
t6 =BinTree( -20 , t9 , t2 );
t4 = t7 .right();
t4 =BinTree( -6 , t8 , t1 );
t9 =BinTree( 8 , t3 , t6 );
t2 =BinTree( -18 , t1 , t0 );
t1 =BinTree( 9 , t0 , t8 );
t6 =BinTree( 15 , t4 , t6 );
t8 =BinTree( -13 , t6 , t2 );
t7 =BinTree( 7 , t2 , t4 );
cout<<sum( t6 )<<endl;
t9 =BinTree( 18 , t0 , t8 );
t1 =BinTree( -4 , t1 , t0 );
t4 = t0 .left();
t1 =BinTree( -12 , t9 , t6 );
t3 =BinTree( -15 , t8 , t0 );
cout<< t6 <<endl;
cout<< t4 <<endl;
t4 =BinTree( 0 , t6 , t2 );
cout<<size( t7 )<<endl;
t9 =BinTree( -7 , t8 , t7 );
t2 =BinTree( -2 , t9 , t9 );
t2 =BinTree( 9 , t7 , t6 );
cout<<sum( t3 )<<endl;
cout<< t1 <<endl;
t9 =BinTree( -6 , t1 , t4 );
t1 = t0 .left();
t8 =BinTree( -7 , t7 , t0 );
t8 = t0 .right();
cout<<sum( t2 )<<endl;
t6 = t1 .right();
t2 =BinTree( -4 , t2 , t2 );
cout<< t5 <<endl;
t9 =BinTree( 9 , t0 , t2 );
cout<< t3 <<endl;
t0 = t4 .right();
t9 = t1 .right();
cout<<size( t6 )<<endl;
cout<< t5 <<endl;
cout<< t4 <<endl;
t3 =BinTree( -18 , t6 , t0 );
t2 =BinTree( 1 , t9 , t4 );
cout<< t5 <<endl;
cout<<sum( t7 )<<endl;
t1 =BinTree( -10 , t4 , t6 );
t4 = t2 .left();
cout<<size( t6 )<<endl;
cout<< t1 <<endl;
cout<< t8 <<endl;
cout<< t9 <<endl;
t8 = t3 .right();
t3 = t8 .right();
t7 =BinTree( 6 , t6 , t8 );
t6 =BinTree( 20 , t8 , t7 );
t3 =BinTree( 9 , t3 , t3 );
t2 =BinTree( 18 , t6 , t5 );
```

```
cout<<sum( t6 )<<endl;
t7 =BinTree( 16 , t9 , t6 );
t1 =BinTree( 9 , t0 , t4 );
t0 =BinTree( -19 , t8 , t6 );
t4 = t7 .left();
t9 =BinTree( 20 , t2 , t8 );
t0 = t1 .left();
t8 = t9 .right();
cout<< t5 <<endl;
t2 =BinTree( 11 , t5 , t7 );
t9 =BinTree( -7 , t4 , t5 );
cout<< t3 <<endl;
t7 = t0 .right();
t1 = t9 .left();
cout<<sum( t7 )<<endl;
cout<<size( t2 )<<endl;
cout<< t3 <<endl;
t9 = t0 .right();
t1 =BinTree( 13 , t9 , t6 );
cout<< t1 <<endl;
cout<<sum( t0 )<<endl;
cout<<sum( t3 )<<endl;
t1 = t9 .left();
t6 =BinTree( 14 , t7 , t7 );
cout<<size( t0 )<<endl;
t7 =BinTree( 6 , t4 , t8 );
t2 =BinTree( -14 , t6 , t0 );
t6 = t4 .right();
t8 =BinTree( 7 , t9 , t6 );
cout<< t7 <<endl;
t0 =BinTree( -15 , t1 , t3 );
t7 = t2 .right();
cout<< t4 <<endl;
t0 =BinTree( 16 , t2 , t9 );
cout<<sum( t9 )<<endl;
t4 = t5 .right();
t3 =BinTree( 19 , t7 , t8 );
cout<<size( t6 )<<endl;
t6 =BinTree( -7 , t7 , t6 );
t7 =BinTree( 0 , t5 , t9 );
cout<<size( t8 )<<endl;
t3 =BinTree( -7 , t8 , t3 );
t3 =BinTree( -7 , t7 , t8 );
t1 =BinTree( 1 , t9 , t0 );
cout<< t8 <<endl;
cout<<sum( t5 )<<endl;
t4 = t3 .right();
cout<< t1 <<endl;
t0 =BinTree( -18 , t9 , t7 );
cout<<size( t9 )<<endl;
cout<< t2 <<endl;
cout<<sum( t3 )<<endl;
t2 =BinTree( -2 , t2 , t0 );
cout<<size( t1 )<<endl;
cout<< t5 <<endl;
t4 =BinTree( 9 , t1 , t7 );
t1 = t1 .right();
t4 =BinTree( -3 , t0 , t8 );
t1 = t6 .left();
t1 =BinTree( 3 , t9 , t6 );
t8 = t0 .left();
t1 = t0 .left();
t9 =BinTree( 12 , t7 , t5 );
t4 = t6 .right();
t9 =BinTree( -9 , t7 , t4 );
cout<<size( t1 )<<endl;
cout<< t0 <<endl;
t8 =BinTree( 19 , t5 , t3 );
cout<<sum( t4 )<<endl;
t5 =BinTree( -12 , t4 , t9 );
cout<<sum( t3 )<<endl;
t2 =BinTree( -16 , t2 , t2 );
t3 =BinTree( 13 , t4 , t6 );
t3 =BinTree( 20 , t3 , t4 );
t3 = t9 .right();
cout<< t2 <<endl;
t8 = t9 .left();
t4 =BinTree( -10 , t7 , t6 );
cout<<size( t7 )<<endl;
cout<<sum( t3 )<<endl;
t3 = t8 .left();
cout<< t0 <<endl;
cout<< t1 <<endl;
cout<< t2 <<endl;
cout<< t3 <<endl;
cout<< t4 <<endl;
cout<< t5 <<endl;
cout<< t6 <<endl;
cout<< t7 <<endl;
cout<< t8 <<endl;
cout<< t9 <<endl;
```

Exemple de sortida 3

```
()
()
1
0
-1
13(,-15(,7))
0
()
0
0
()
-1
-1
-15(,7)
-15(,7)
-1
()
3
-8
14
24
0
()
24
20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7)))
60
23
-11(18(-15(,7),-15(,7)),)
58
15(-6(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))),20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7)))),-
-10(,-1)
39
0
-12(18(-7(-10(,-1),),-13(15(-6(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))),20(18(-15(,7),-15(,7)
104
-11(18(-15(,7),-15(,7)),)
-15(-13(15(-6(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))),20(18(-15(,7),-15(,7)),18(-15(,7),-15(
1
-11(18(-15(,7),-15(,7)),)
0(15(-6(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))),20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))))
-11(18(-15(,7),-15(,7)),)
37
```

```
1
-10(0(15(-6(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))
()
-1
1
-11(18(-15(,7),-15(,7)),)
9(-7(-10(,-1),),-7(-10(,-1),))
-18
42
9(-7(-10(,-1),),-7(-10(,-1),))
13(-7(-10(,-1),),20(-18(20(18(-15(,7),-15(,7)),18(-15(
-12
-27
15
6(-1,-18(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))),-
-1
-18
0
4
7(-7(-10(,-1),),)
-9
1(-7(-10(,-1),),16(-14(14(-7(-10(,-1),),-7(-10(,-1),))
3
-14(14(-7(-10(,-1),),-7(-10(,-1),)),-18(20(18(-15(,7),
-45
31
-11(18(-15(,7),-15(,7)),)
3
-18(-7(-10(,-1),),0(-11(18(-15(,7),-15(,7)),),-7(-10(,
0
-45
-16(-2(-14(14(-7(-10(,-1),),-7(-10(,-1),)),-18(20(18(-
10
0
-18(-7(-10(,-1),),0(-11(18(-15(,7),-15(,7)),),-7(-10(,
-7(-10(,-1),)
-16(-2(-14(14(-7(-10(,-1),),-7(-10(,-1),)),-18(20(18(-
-11(18(-15(,7),-15(,7)),)
-10(0(-11(18(-15(,7),-15(,7)),),-7(-10(,-1),)),-7(-18(
-12(,-9(0(-11(18(-15(,7),-15(,7)),),-7(-10(,-1),)),))
-7(-18(20(18(-15(,7),-15(,7)),18(-15(,7),-15(,7))),-7(
0(-11(18(-15(,7),-15(,7)),),-7(-10(,-1),))
0(-11(18(-15(,7),-15(,7)),),-7(-10(,-1),))
-9(0(-11(18(-15(,7),-15(,7)),),-7(-10(,-1),)),)
```

Observació

La solució d'aquest exercici s'ha de basar en un ús raonable del tipus `BinTree`. Qualsevol solució que ignori això i faci servir enfocaments o estructures de dades alternatives que no formen part de l'assignatura serà invalidada.
Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, on cada operació té cost **CONSTANT** (excepte per a la d'escriptura d'arbres, que s'espera cost proporcional a la mida de l'arbre involucrat), i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autor : PRO2
Generació : 2024-04-09 15:25:54