
Arbre general és un arbre compta graus?**X54738_ca**

Donada la classe *Arbre* que permet gestionar arbres generals usant memòria dinàmica, cal implementar el mètode

```
bool es_arbre_compta_graus ();
```

que comprova que el contingut de cada node coincideix amb el seu grau.

Cal enviar a jutge.org la següent especificació de la classe *Arbre* i la implementació del mètode dins del mateix fitxer. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre d'elements n de l'arbre.

```
#include <cstdlib>
#include <string>
using namespace std;
typedef unsigned int nat;
```

```
template <typename T>
class Arbre {
```

```
public:
```

```
// Construeix un Arbre format per un únic node que conté a x.
```

```
Arbre(const T &x);
```

```
// Tres grans.
```

```
Arbre(const Arbre<T> &a);
```

```
Arbre& operator=(const Arbre<T> &a);
```

```
~Arbre() throw();
```

```
// Col·loca l'Arbre donat com a primer fill de l'arrel de l'arbre sobre el que s'aplica el
mètode i l'arbre a queda invalidat; després de fer b.afegir_fill(a), a no és un arbre vàlid.
```

```
void afegir_fill (Arbre<T> &a);
```

```
static const int ArbreInvalid = 400;
```

```
// Comprova que el contingut de cada node coincideix amb el seu grau
```

```
bool es_arbre_compta_graus () const;
```

```
private:
```

```
Arbre(): _arrel (NULL) {};
```

```
struct node {
```

```
    T info;
```

```
    node* primf;
```

```
    node* seggerm;
```

```
};
```

```
node* _arrel ;
```

```
static node* copia_arbre (node* p);
```

```
static void destrueix_arbre (node* p) throw();
```

```
// Aquí va l'especificació dels mètodes privats addicionals  
};
```

```
// Aquí va la implementació del mètode es_arbre_compta_graus
```

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *Arbre* i un programa principal que llegeix un arbre general i després crida el mètode *es_arbre_compta_graus*.

Entrada

L'entrada consisteix en la descripció d'un arbre general d'enters (el seu recorregut en preordre, en el qual al valor de cada node li segueix el seu nombre de fills).

Sortida

Una línia amb el text "NO és arbre compta graus" indicant que l'arbre no és compta graus o amb el text "SI és arbre compta graus" indicant que l'arbre si és compta graus.

Observació

Només cal enviar la classe requerida i la implementació del mètode *es_arbre_compta_graus* amb el seu cost en funció del nombre d'elements *n* de l'arbre. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

Exemple d'entrada 1

```
2 2  
 1 1  
  1 1  
   3 3  
    0 0  
    0 0  
    0 0  
  2 2  
   1 1  
    5 5  
     1 1  
      0 0  
      0 0  
      3 3  
       0 0  
       0 0  
       0 0  
       0 0  
      2 2  
       0 0  
       0 0  
0 0
```

Exemple d'entrada 2

```
2 2  
 1 1  
  1 1  
   3 3
```

Exemple de sortida 1

```
SI és arbre compta graus
```

```
0 0  
0 0  
0 0  
2 2  
 1 1
```

```
6 5
 1 1
  0 0
 0 0
 3 3
  0 0
  0 0
  0 0
 0 0
 2 2
  0 0
  0 0
0 0
```

Exemple d'entrada 3

```
7 0
```

Exemple d'entrada 4

```
1 1
 0 0
```

Informació del problema

Autor : Jordi Esteve

Generació : 2025-10-21 20:48:16

© *Jutge.org*, 2006–2025.

<https://jutge.org>

Exemple de sortida 2

NO és arbre compta graus

Exemple de sortida 3

NO és arbre compta graus

Exemple de sortida 4

SI és arbre compta graus