
Arbre general és un arbre suma?

X88982_ca

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

```
bool es_arbre_suma();
```

que comprova que el contingut de cada node coincideix amb la suma dels nodes dels seus fills, exceptuant les fulles (els de grau 0).

Cal enviar a jutge.org la següent especificació de la classe *Arbre* i la implementació del mètode dins del mateix fitxer.

```
#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 la suma dels nodes
```

```
    // dels seus fills, exceptuant les fulles (els de grau 0).
```

```
    bool es_arbre_suma() 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_suma

```

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_suma*.

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 suma" indicant que l'arbre no és un arbre suma o amb el text "SI és arbre suma" indicant que l'arbre si és un arbre suma.

Observació

Només cal enviar la classe requerida i la implementació del mètode *es_arbre_suma*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

Exemple d'entrada 1

```

42 2
  -5 1
    -5 1
      -5 3
        1 0
        2 0
      -8 0
47 2
  41 1
    41 5
      2 1
      2 0
      7 0
    21 3
      8 0
      9 0
      4 0
    3 0
    8 2
      1 0
      7 0
6 0

```

Exemple d'entrada 2

```

42 2
  -5 1

```

Exemple de sortida 1

```

SI és arbre suma

```

```

-5 1
  -5 3
    1 0
    2 0

```

```
      -8 0
47 2
  41 1
    20 5
      2 1
        2 0
          7 0
            21 3
              8 0
                9 0
                  4 0
                    3 0
                      8 2
                        1 0
                          7 0
6 0
```

Exemple d'entrada 3

```
7 0
```

Exemple d'entrada 4

```
0 2
3 0
-3 0
```

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T17:04:20.029Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>

Exemple de sortida 2

```
NO és arbre suma
```

Exemple de sortida 3

```
SI és arbre suma
```

Exemple de sortida 4

```
SI és arbre suma
```